

 Open access • Journal Article • DOI:10.1049/IP-I-2.1989.0059

## Fast search algorithm for VQ-based recognition of isolated words — [Source link](#)

Sin-Horng Chen, J. S. Pan

**Institutions:** National Chiao Tung University

**Published on:** 01 Dec 1989

**Topics:** Search algorithm, Feature vector and Codebook

Related papers:

- [An Improvement of the Minimum Distortion Encoding Algorithm for Vector Quantization](#)
- [An Algorithm for Vector Quantizer Design](#)
- [An algorithm for finding nearest neighbours in \(approximately\) constant average time](#)
- [Vector Quantization and Signal Compression](#)
- [Vector quantization](#)

Share this paper:    

View more about this paper here: <https://typeset.io/papers/fast-search-algorithm-for-vq-based-recognition-of-isolated-3mufog2qce>

# Fast search algorithm for VQ-based recognition of isolated words

S.H. Chen  
J.S. Pan

*Indexing term:* Algorithms

**Abstract:** This paper presents a fast search algorithm for vector quantisation (VQ)-based recognition of isolated words. It incorporates the property of high correlation between speech feature vectors of consecutive frames with the method of triangular inequality elimination to relieve the computational burden of vector-quantising the test feature vectors by full codebook search, and uses the extended partial distortion method to compress the incomplete matching computations of wildly mismatched words. Overall computational load can therefore be drastically reduced while the recognition performance of full search can be retained. Experimental results show that about 93% of multiplications and additions can be saved with a little increase of both comparisons and memory space.

## 1 Introduction

Many methods have been proposed in the past few years for isolated-word recognition [1–3]. Among them the most commonly used techniques include dynamic time warping (DTW) [4], vector quantisation (VQ) [5], and hidden Markov model (HMM) [6]. Dynamic time warping is a nonparametric method which uses dynamic programming algorithm to find the best matched word for the test utterance. For isolated word recognition, DTW usually has good performance but with high computational complexity. The hidden Markov model is a method which uses a probabilistic model to describe the acoustic characteristics of a speech signal. It can be applied to isolated word recognition in both speaker dependent and speaker independent cases with good results. But the computational complexities are very high for both training and test processes. Vector quantisation firstly uses a complicated recursive algorithm to find representative codewords for each word, and then finds the best matched word as the recognised result by full search [5]. Considering both computation efficiency and recognition performance, VQ is a promising method for isolated word recognition.

In VQ method, a codebook of a fixed size is trained for each word by using a large set of training samples. In

the test, feature vectors of the test utterance are independently vector-quantised by using full search to find the codeword with minimum distortion in each codebook. The quantisation errors are accumulated and taken as the dissimilarity measure between the test utterance and the word associated to the codebook. The word with smallest dissimilarity measure is decided to be the recognised word.

Usually a full search algorithm is applied in VQ speech recognition, but it is a time consuming process when the codebook size is large. Many methods were proposed to solve this problem. For example, binary tree search VQ [18] was proposed to logarithmically reduce the number of operations. But the performance of using binary tree search VQ is usually degraded. Multisection VQ [7] was proposed by Burton *et al.* for isolated word recognition. It segments the utterance of a word into several sections and uses a codebook of smaller size for each section. The computation time can be saved because codebooks of smaller size are used. However, the performance of recognition is sensitive to the segmentation of the utterance of a word into several sections. Finite state VQ was introduced to adaptively switch the codebook based on the change of state such that codebooks of smaller size can be used [18]. However, the design of state transition function is a relatively difficult job. In this paper, we stay with the full search VQ and solve this problem by finding fast computation algorithm.

To find the best matched codeword from a codebook is a pattern matching problem. Many fast algorithms have been studied in the past. Fischer and Patrick [17] proposed a preprocessing algorithm to reorder the design samples such that many distance computations could be eliminated. Fukunaga and Narendar [8] proposed a branch and bound algorithm (BAB) for computing  $k$  nearest neighbours. A branch and bound algorithm is a tree search algorithm using a hierarchical decomposition of the sample set of known patterns. They used the criterion of triangular inequality to develop two rules to eliminate the distance computation in the tree classifier. Kamgar-Parsi and Narendar [9] added another two rules to the BAB algorithm to improve the computation time. Niemann and Goppert [11] combined these four rules into one and used hierarchical partition of pattern sample algorithm to get more efficient computation time. Bei and Gray [12] proposed an algorithm to minimise the computation time by using the same idea as the 'partial distance method' [14]. Vidal [13] presented the approximating and eliminating search algorithm (AESA) in which the computation time is approximately constant for codeword search in a large size codebook. There are many other high speed searching algorithms for vector quantisation [14–16]. All algorithms mentioned above

Paper 69331 (E7, E8, E16), first received 10th January and in revised form 9th August 1989

The authors are with the Department of Communication Engineering and Centre for Telecommunications Research, National Chiao Tung University, Hsinchu, Taiwan, Republic of China.

are by no means using the correlation relationship among input data vectors.

In this paper we propose a simple and efficient full search VQ algorithm for isolated word recognition. The basic idea is the same as the finite state VQ which uses the property of high correlation between contiguous speech signals to relieve the computational burden of codeword search. This algorithm incorporates the correlation property of speech signal to the elimination rule based on triangular inequality for encoding test feature vectors, and uses a modified partial distortion method to control the process of computing the cumulative distortions of the test utterance to all candidate words. We can therefore expect that the computations of frame distortions for the recognised word will be very fast because many codeword searches can be eliminated due to the high correlation between contiguous frames. The computations of cumulative distortions for other words will also be very fast because it can terminate the computations of wildly mismatched words as soon as possible.

## 2 Fast VQ search algorithm for isolated-word recognition

In a VQ isolated word recognition system, a codebook for each word is generated first from training data. Then, the test utterance is encoded frame by frame using the codebook. The accumulated encoding error is used as the distortion measure between the test utterance and the word associated to the codebook. Recognition result is obtained by finding the word with minimum distortion. For a conventional full search VQ, the computational complexity is evaluated as follows:

$$\begin{aligned} \text{Multiplication number} &= T \cdot CB \cdot N \cdot VD \\ \text{Addition number} &= (2 \cdot VD - 1) \cdot N \cdot T \cdot CB \\ &\quad + (T - WN) \cdot CB \\ &= 2 \times \text{Multiplication number} \\ \text{Comparison number} &= CB \cdot T \cdot (N - 1) \\ &\quad + (CB - 1) \cdot WN \end{aligned}$$

where  $T$  is the total frame number of all test utterances,  $N$  is the size of codebook,  $CB$  is the codebook number,  $VD$  is the dimension of feature vector, and  $WN$  is the number of test utterances.

Because the computational complexity of a conventional method is very high, many methods [7-17] were proposed to relieve the computational burden. Among them, triangular inequality elimination [8, 13] is the most popular one. One form of triangular inequality elimination (TIE) is presented below.

Let  $\text{Dis}(X, C_1)$  be the distance between test vector  $X$  and codeword  $C_1$ . If  $\text{Dis}(C_1, C_2) > 2 \text{Dis}(X, C_1)$ , then eliminate the computation of  $\text{Dis}(X, C_2)$  because it is always greater than  $\text{Dis}(X, C_1)$ . For square-error distortion measure, the condition of elimination can be changed to  $D(C_1, C_2) > 4 D(X, C_1)$  such that the square root operation in  $\text{Dis}(\cdot, \cdot)$  is not needed.

Although many other triangular inequality elimination rules exist [13], the one stated above is the simplest and very efficient one if we can find at the beginning a codeword which has very small distance (distortion) to the test vector. This is because a lot of distortion computations can therefore be eliminated. One way to find a good

candidate of codeword at the beginning of the encoding process of a new test vector is to use the correlation relationship existing among contiguous test vectors. Because high correlation exists between vectors of two consecutive frames of speech signal, the encoded result of the previous frame is helpful to find the best codeword to be first calculated for the current frame. That means the codeword recommended by the encoded result of previous frame will have very small distortion to the test vector of current frame with high probability. The codeword search for the test vector of current frame that starts from this best codeword can therefore get some computational gains by applying the triangular inequality elimination.

A simple realization of this idea is to empirically construct from the training utterances a transition table whose  $i$ th row indicates the index of the encoded codeword of the current frame vector sorted in descending order according to the occurrence probability when the previous frame vector is encoded as the  $i$ th codeword. From this table, the best codeword for starting the encoding of the current frame vector can be obtained by finding the first element of the  $i$ th row if the encoded codeword of the previous frame is given to be the  $i$ th codeword. For the initial frame, the statistics of the encoded results of the first frame of the training utterances can be used to build an initial table. In order that the TIE can be applied, one distortion table whose  $(i, j)$  element indicates the distortion between the  $i$ th and the  $j$ th codewords must also be built for each codebook. And every row of each distortion table should be sorted in its increasing order such that the TIE can be efficiently applied.

Based on the idea stated above, we present in the following a fast algorithm of codeword searching in the codebook of a word for test utterances. The procedure of this algorithm is listed below.

*Step 1:* For the first frame, set  $i = 1$ , look up the initial table to get the most probable codeword  $j$ , and compute  $D(X_i, C_j)$ . Then set  $D_{\min} = D(X_i, C_j)$  and let  $l = j$ . Here  $X_i$  is the feature vector of the  $i$ th frame in the test utterance,  $C_j$  is the  $j$ th codeword, and  $D(\cdot, \cdot)$  is the square error distortion measure.

*Step 2:* Use TIE to eliminate all impossible codewords based on the comparisons of  $4 \cdot D_{\min}$  with all elements of the  $l$ th row of the distortion table.

*Step 3:* If no codewords are left, then the codeword is found, set  $i = i + 1$  and go to step 4; otherwise, look up the initial table to get the next probable codeword  $C_k$ , compute  $D(X_i, C_k)$ , set  $D_{\min} = \min [D_{\min}, D(X_i, C_k)]$ , let  $l = k$ , and go to step 2.

*Step 4:* Find the best codeword  $C_m$  (i.e. the first element) from the  $p$ th row of the transition table to start the encoding of the  $i$ th frame vector based on the knowledge of the encoded result of the  $(i-1)$ th frame to be the codeword  $C_p$ . Then compute  $D(X_i, C_m)$ , set  $D_{\min} = D(X_i, C_m)$ , and let  $l = m$ .

*Step 5:* Use TIE to eliminate all impossible codewords by comparing  $4 \cdot D_{\min}$  with all elements of the  $l$ th row of the distortion table.

*Step 6:* If no codewords are left, then encoded codeword for this frame is found, check whether it is the last frame or not. If it is, then terminate the algorithm; else set  $i = i + 1$  and go to step 4. Otherwise, get the next probable codeword  $C_n$  (i.e., the next element) from the  $p$ th row of the transition table, calculate  $D(X_i, C_n)$ , set  $D_{\min} = \min [D_{\min}, D(X_i, C_n)]$ , let  $l = n$ , and go to step 5.

This algorithm can be simplified by further considering the characteristic of short term stationarity of speech signal. Because speech waveform is a slowly varying signal, the acoustic features of speech signal are usually not quickly changed from one frame to the next. We can therefore announce that the most probable codeword to encode the current frame is the same codeword that encodes the previous frame, and the next probable codeword is the one with smallest distortion to the encoded codeword of the previous frame, and so on. A simple sub-optimum search algorithm is accordingly introduced by modifying the original fast search algorithm. First, the distortion tables are generated as before. Every row of each distortion table is also sorted into increasing order. Then, the transition table is generated from the sorted distortion table by replacing all distortion elements with the indices of codewords associated to them. The same codeword search procedure discussed before is then applied.

Partial distortion method [12] was originally introduced to reduce the computation of codeword search in a codebook. It is easy to incorporate partial distortion method to the above fast codeword search algorithm by simply giving up the unfinished codeword distortion computation if the partial distortion is already larger than a complete codeword distortion.

After considering the efficiency of codeword searching in a codebook, the overall computation efficiency can be further improved by abandoning the computations of cumulative distortions of the wildly mismatched words. A modified partial distortion method is introduced here for this purpose. First, we consider the case of directly applying the partial distortion method to the word recognition. The partial cumulative distortion of encoding the test utterance by using a codebook of some word is computed and continuously compared with the current minimum total cumulative distortion. If it is larger, then the word is abandoned and the next one is considered. Otherwise, we continued to compute the partial cumulative distortion and to compare it with the minimum total cumulative distortion until the total cumulative distortion was obtained. After a new total cumulative distortion was computed, the minimum total cumulative distortion was updated by comparing it with the current minimum one and taking the minimum of them. This procedure is continued until all words are processed and the recognised word decided upon.

Although the above direct application of partial distortion method to word recognition is a simple and effective method, a modified version called extended partial distortion method is more efficient for compressing the unnecessary computations of cumulative distortions in isolated word recognition. The procedure of this algorithm is stated as follows:

*Step 1:* Let  $l_i = 1$  and calculate the distortion  $D_i$  between  $X_1$  and the  $i$ th codebook, for  $i = 1, \dots, CB$ . Here  $X_1$  is the feature vector of the first frame of input signal and  $CB$  is the number of codebook.

*Step 2:* Find  $D_s = \min_i D_i$  and  $s = \arg \min_i D_i$ .

*Step 3:* If  $l_s = T$ , then set the  $s$ th word to be the recognised word and terminate the algorithm; otherwise, set  $l_s = l_s + 1$ , calculate the encoding distortion of  $X_{l_s}$  by using the  $s$ th codebook and add it to  $D_s$ , and go to step 2. Here  $T$  is the frame number of the test utterance.

We can incorporate the partial distortion method with the fast codeword search algorithm and then combine it

with the extended partial distortion method and use it in isolated word recognition. This algorithm uses the extended partial distortion method in codebook's cumulative distortion computation to abandon incomplete wildly mismatched words, and uses both triangular inequality elimination and partial distortion method to get rid of all impossible codewords. It is a fast VQ search algorithm for word recognition. The procedure is described as follows:

*Step 1:* Let  $l_i = 1$  and calculate the distortion  $D_i$  of  $X_1$  by using the  $i$ th codebook, for  $i = 1$  to  $CB$ . Here  $X_1$  is the feature vector of the first frame of the test utterance. The distortion  $D_i$  is calculated as follows:

*Step 1a:* For the frame  $l_i$ , look up the initial table of word  $i$  to get the most probable codeword  $j$ , compute  $D(X_{l_i}, C_j^i)$ , and set  $D_i = D(X_{l_i}, C_j^i)$ . Here  $C_j^i$  is the  $j$ th codeword of codebook  $i$ .

*Step 1b:* Use TIE to eliminate all impossible codewords.

*Step 1c:* If no codewords are left, then terminate the computation; otherwise, go to Step 1d

*Step 1d:* Find the next probable codeword  $m$  and compute  $D(X_{l_i}, C_m^i)$ . The partial distortion method is used in the computation of  $D(X_{l_i}, C_m^i)$  to eliminate unfinished distortion computation of impossible codeword. If  $D(X_{l_i}, C_m^i)$  is an unfinished distortion, then go to Step 1c; otherwise, set  $D_i = \min [D_i, D(X_{l_i}, C_m^i)]$  and go to Step 1b

*Step 2:* Find  $D_s = \min_i D_i$  and  $s = \arg \min_i D_i$ .

*Step 3:* If  $l_s = T$ , then set the  $s$ th word to be the recognised word and terminate the algorithm; otherwise, set  $l_s = l_s + 1$ , calculate the distortion of  $X_{l_s}$  to the  $s$ th codebook, add it to  $D_s$ , and go to Step 2. The distortion computation follows the same procedure of Steps 1a-1d except that the transition table is substituted for the initial table.

### 3 Simulations

Several experiments were done to examine the effectiveness of the proposed Fast VQ Recognition Algorithm for speaker-independent isolated-word recognition. The system is similar to a conventional VQ-based recognition system except that the full-search encoding of feature vector is replaced by the proposed fast search algorithm. The speech signal is firstly recorded using a microphone and is low pass filtered from 0 to 4 kHz. Then it is digitised into 12 bits at a sampling frequency of 10 kHz. Twelve LPC-derived cepstral coefficients are then calculated and taken as a feature vector for each 25.6 ms frame spaced every 12.8 ms. After extracting feature vectors, two phases are included in the recognition process. They are the training and the test phases. In the training phase, binary codeword splitting and LBG recursive algorithm are used to generate one codebook for each word. Each codebook contains 16 codewords. In the test phase, the cumulative distortions of encoding the test utterance using these codebooks are calculated. The word with the smallest cumulative distortion is regarded as the recognised word.

The vocabulary contains 21 Mandarin words, all of which are monosyllables. The training data consists of utterances of twenty repetitions of these 21 words collected from ten male speakers. Utterances of five repetitions generated by another five male speakers were taken as the test data. All the recordings were done in a computer room. The total frame number of test data is 3437.

In all the following experiments, the same recognition rate of 86.67% was achieved for the outside test. It is a little worse than that (88% for the test of a 20 word vocabulary) obtained in [5]. This little performance degradation is mainly owing to the fact that several confusable word pairs do exist in our 21 word vocabulary.

To perform our fast recognition algorithm, several transition tables were generated from the training data. First, one transition table for each codebook was established by using all training utterances and referred to as total state transition (TST) table. Then another transition table was established for each codebook by using only the training utterances of the same word associated to the codebook and referred to as state transition (ST) table. One distortion table of all pairs of codewords was established for each codebook. It is sorted into increasing order for each row. The transition table generated from the distortion table is referred to as the previous codeword transition (PCT) table.

First we examined the efficiency of incorporating the partial distortion method with the fast codeword search algorithm for encoding the vectors of test utterances. It is called a TST-type if the TST transition table is used. It is ST-type when the ST transition table is used. For simplified suboptimum version, the PCT transition table is used and is, therefore, called as PCT-type. The conventional full codebook search method is referred to as the conventional-(CVT)-type. For each utterance, test feature vectors were encoded by using the codebook of the same word associated to it.

Table 1 lists the experimental results. It is noted that because the additional number is almost equal to two

**Table 1: Number of operations of encoding test feature vectors**

Type	Operations	
	multiplication	comparison
CVT	659904	51555
TST	112691	136265
ST	107149	127714
PCT	107158	128035

Each utterance is encoded by using the codebook of the same word associated to it.

times the multiplication number for all types, we do not count it in our experiment. From Table 1, it can be seen that the multiplication number is much less for all three types of the proposed algorithm when compared with the CVT-type. Approximately 83% of multiplications can be saved. However, about two times as many comparisons are needed. The ST-type was found to be slightly faster. The PCT-type produced nearly the same performance as the ST-type.

The elimination efficiency of TIE in the fast codeword search algorithm is then checked. It contains two cases. The first case is to test the elimination efficiency of TIE when each test utterance is encoded by using the codebook of the same word associated to it. The second case is to examine the elimination efficiency of TIE when all codebooks were used to encode the test utterances. Both cases emphasise the examination of the effects of applying TIE on the first several codeword eliminations. This can help us determine the number of eliminations to be used in TIE with the greatest efficiency.

Tables 2 and 3 display the experimental results of the first and the second cases, respectively. As shown in these tables, one codeword elimination is almost good enough in both cases. It can eliminate on average, 8.32, 8.87, and

8.93 codeword distortion computations in the first case for TST-, ST-, and PCT-type, respectively; and 4.83, 5.02, and 5.06 in the second case. The elimination efficiency

**Table 2: Elimination efficiency of fast codeword search algorithm**

No. of TIE elimination operation	Average no. of codewords eliminated		
	TST	ST	PCT
1	8.328	8.866	8.934
2	0.615	0.385	0.357
3	0.253	0.270	0.212
4	0.162	0.151	0.208
5	0.176	0.134	0.146
6	0.111	0.054	0.062
7	0.038	0.050	0.046
8	0.024	0.031	0.051
9	0.032	0.027	0.015
10	0.018	0.018	0.019
11	0.009	0.002	0.002
12	0.007	0.002	0.004
13	0.013	0.001	0.001
14	0.002	0.001	0.001
15	0.000	0.000	0.000

Each utterance is encoded by using the codebook of the same word associated to it.

**Table 3: Elimination efficiency of fast codeword search algorithm**

No. of TIE elimination operation	Average no. of codewords eliminated		
	TST	ST	PCT
1	4.828	5.028	5.061
2	0.334	0.298	0.240
3	0.149	0.180	0.149
4	0.122	0.111	0.137
5	0.086	0.065	0.080
6	0.049	0.044	0.050
7	0.033	0.031	0.028
8	0.015	0.016	0.026
9	0.014	0.014	0.015
10	0.010	0.008	0.010
11	0.005	0.003	0.005
12	0.004	0.001	0.003
13	0.008	0.001	0.001
14	0.008	0.000	0.000
15	0.003	0.000	0.000

Test utterances are encoded by using codebooks of all words.

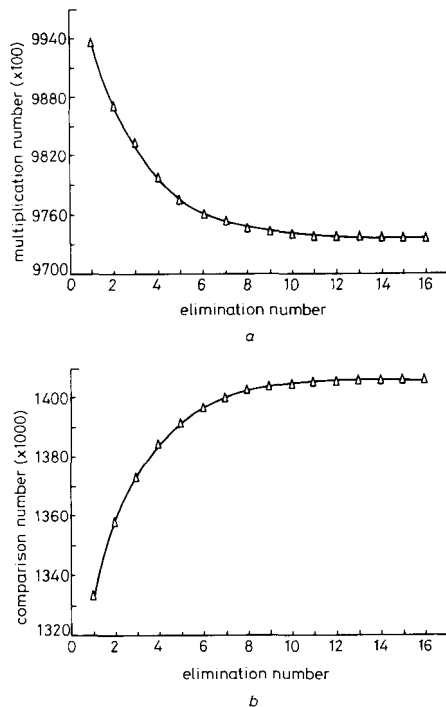
decreases as the number of eliminations increased. Comparing results of these two cases, we found that TIE elimination is more effective if the word associated to the codebook is the same as the test utterance. This observation coincides with our expectation because, in the first case, the codeword recommended by the encoded result of the previous frame is usually a good one, which has very small distortion to the current test vector. In isolated word recognition, the ineffectiveness of applying fast codeword search algorithm in codebooks of wrong words will be greatly compensated by using the extended partial distortion method.

The computational efficiency of the fast VQ recognition algorithm for isolated word recognition is tested. We still call it either the TST-, the ST-, or the PCT-types when TST, ST, or PCT transition tables are used, respectively. The conventional method is called the CVT-type. The experimental results are listed in Table 4. It shows that all three types of fast VQ recognition algorithm can save about 93% of multiplications with a little increase in the comparison operations, compared with the conventional CVT-type.

**Table 4: Computation efficiency of fast VQ recognition algorithm for isolated word recognition**

Type	Operations	
	multiplication ( $\times 10^6$ )	comparison
CVT	13.858	1.085
TST	0.995	1.435
ST	0.967	1.400
PCT	0.974	1.406

Although elimination efficiency shown in Tables 2 and 3 decreases as the number of TIE operations increases, computational savings of both multiplications and additions still progress as more TIE operations are applied. But the rates of computation savings decrease very quickly. However, when more TIE operations are applied, the algorithm becomes more complicated as well making the need for more comparisons. A compromise must be reached. This test is to explore how many times elimination should be used for encoding a test vector by each codebook in fast VQ recognition algorithm. Only a PCT-type algorithm is considered in this study because it is almost as good as the ST-type, but much simpler. The elimination number is used as a parameter to calculate the numbers of operations. The experimental results are drawn in Fig. 1. It shows that when only one elimination

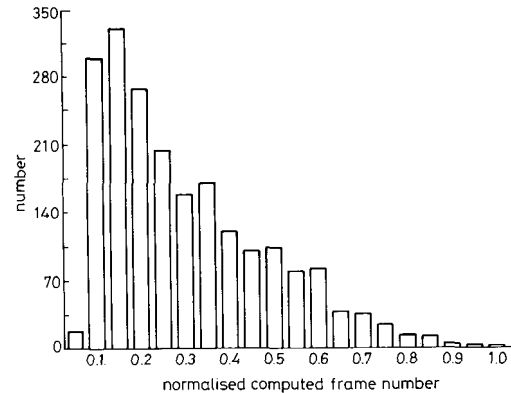


**Fig. 1** Computation efficiency of PCT-type fast VQ recognition algorithm:

a multiplication  
b comparison

is made, almost the same performance can be obtained as doing it 16 times. One additional advantage of using elimination only once is that the transition table is not needed, because we can simply use the encoded codeword of the previous frame as the best codeword for the current frame to make the TIE elimination.

Finally, the performance of applying the extended partial distortion method in computing the encoded cumulative distortions of codebooks is examined. For each test utterance, the extensions of frame number in cumulative distortion computation of unrecognized words are computed and normalised by the frame number of this test utterance. Fig. 2 shows the histogram



**Fig. 2** Histogram of normalised computed frame number of test utterances in isolated-word recognition by using fast VQ recognition algorithm

of the normalised computed frame number of test utterances in the isolated word recognition test using fast VQ recognition algorithm. Twenty subdivided cells are used in the calculation of the histogram. It is noted that the frame extension using the codebook associated to the recognised word is not included because it is always equal to 1. From this figure, on average, only a small portion of computation is accomplished in the calculation of cumulative distortion for unrecognized words. It is also found that most long frame-extension cases are the result of confused words. However, TIE operation works efficiently in those cases. So our fast VQ recognition algorithm that combines the extended partial distortion method and the fast codeword search algorithm is extremely effective in word recognition.

#### 4 Conclusions

In this paper, a fast search algorithm for VQ-based recognition of isolated words was discussed. It uses a high correlation characteristic of speech signal to increase the efficiency of triangular inequality elimination in the VQ codeword search. It also extends the idea of the partial distortion method to compress the unnecessary frame distortion computations. Savings of about 93% on both multiplications and additions were achieved with a little increase in comparisons in our speaker independent isolated word recognition test. Another result of computational saving is that additional memory for transition and distortion tables is needed.

This algorithm can also be applied to other areas, such as image coding, codebook training, speech coding, speaker recognition, etc.

#### 5 Acknowledgment

This work was supported by ITRI, Taiwan, Republic of China.

## 6 References

- 1 RABINER, L.R., and LEVINSON, S.E.: 'Isolated and connected word recognition-theory and selected application', *IEEE Trans., COM-29*, 1981, (5), pp. 621-659
- 2 O'SHAUGHNESSY, D.: 'Speech Communication: Human and machine'. Addison-Wesley, 1987
- 3 O'SHAUGHNESSY, D.: 'Speaker recognition', *IEEE ASSP Magazine*, 1986, pp. 4-17
- 4 SAKOE, H., and CHIBA, S.: 'Dynamic programming algorithm optimization for spoken word recognition', *IEEE Trans., ASSP-26*, (1), 1978, pp. 43-49
- 5 SHORE, J.E., and BURTON, D.K.: 'Discrete utterance speech recognition without time alignment', *IEEE Trans., IT-29*, (4), 1983, pp. 473-491
- 6 RABINER, L.R., and JUANG, B.H.: 'An introduction to hidden Markov models', *IEEE ASSP Magazine*, 1986, pp. 4-16
- 7 BURTON, D.K., SHORE, J.E., and BUCK, J.T.: 'Isolated-word speech recognition using multi-section vector quantization codebooks', *IEEE Trans., ASSP-33*, (4), 1985, pp. 837-849
- 8 FUKUNAGA, K., and NARENDRA, P.M.: 'A branch and bound algorithm for computing k-nearest neighbours', *IEEE Trans., C-24*, 1975, pp. 750-753
- 9 KAMGAR-PARSI, B., and KANAL, L.N.: 'An improved branch and bound algorithm for computing k-nearest neighbours', *Pattern Recog. Lett.*, 3, 1985, pp. 7-12
- 10 LARSEN, S., and KANAL, L.N.: 'Analysis of k-nearest neighbour branch and bound rules', *Pattern Recog. Lett.*, 4, 1986, pp. 71-77
- 11 NIEMANN, H., and GOPPERT, R.: 'An efficient branch-and-bound nearest neighbour classifier', *Pattern Recog. Lett.*, 7, 1988, pp. 67-72
- 12 BEI, C.-D., and GRAY, R.M.: 'An improvement of the minimum distortion encoding algorithm for vector quantization', *IEEE Trans., COM-33*, (10), 1985, pp. 1132-1133
- 13 VIDAL, E.: 'An algorithm for finding nearest neighbours in (approximately) constant average time', *Pattern Recog. Lett.*, 4, 1986, pp. 145-157
- 14 CHENG, D.-Y., GERSHO, A., RAMAMURTHI, B., and SHOHAM, Y.: 'Fast search algorithms for vector quantization and pattern matching', *IEEE ICASSP*, 1984, pp. 9.11.1-4
- 15 CHENG, D.-Y., and GERSHO, A.: 'A fast codebook search algorithm for nearest-neighbour pattern matching', *IEEE ICASSP*, 1986, pp. 6.14.1-4
- 16 SOLEYMANI, M.R., and MORGERAI, S.D.: 'A high-speed search algorithm for vector quantization', *IEEE ICASSP*, 1987, pp. 45.6.1-3
- 17 FISCHER, F.P., and PATRICK, E.A.: 'A preprocessing algorithm for nearest neighbour decision rules', *Proc. Nat. Electronics Conf.*, 26, 1970, pp. 481-485
- 18 GRAY, R.M.: 'Vector quantization', *IEEE ASSP Magazine*, 1984, pp. 4-28