LETTER

A Method for Improving TIE-Based VQ Encoding Introducing RI Rules

Chi-Jung HUANG[†], *Nonmember*, Shaw-Hwa HWANG[†a)], *and* Cheng-Yu YEH[††], *Members*

SUMMARY This study proposes an improvement to the Triangular Inequality Elimination (TIE) algorithm for vector quantization (VQ). The proposed approach uses recursive and intersection (RI) rules to compensate and enhance the TIE algorithm. The recursive rule changes reference codewords dynamically and produces the smallest candidate group. The intersection rule removes redundant codewords from these candidate groups. The RI-TIE approach avoids over-reliance on the continuity of the input signal. This study tests the contribution of the RI rules using the VQ-based, G.729 standard LSP encoder and some classic images. Results show that the RI rules perform excellently in the TIE algorithm.

key words: *G729, Triangular Inequality Elimination, vector quantization*

1. Introduction

Vector quantization (VQ) is a powerful method for image compression because of its excellent rate-distortion performance and simple structure. An algorithm applies a full-search method producing the best-matched codeword. However, the computation requirements of a full-search algorithm are large. Previous research [1] applied triangular inequality elimination (TIE) to VQ-based image coding, achieving more than 90% in computation saving. But it is not an effective method for VQ-based audio coding. The TIE approach depends heavily on the continuous property of the input signal. The definition of the TIE approach is

$$\text{TIE-1: } \{c_t \,|\, d(c_j, c_t) < 2d(c_j, x)\} \tag{1}$$

The reference codeword $\boldsymbol{C}_j$ which is closest to input vector $\boldsymbol{x}$ makes the search space $\{\boldsymbol{C}_t\}$ small. However, the noisy input vector greatly reduces the performance of the TIE approach.

A more sophisticated TIE was proposed to enhance the TIE approach [2], [3]. The definition of this novel approach is

$$\begin{aligned}\text{TIE-2: } \{c_t \,|\, & d(c_j, c_t) < d(c_j, x) + d(c_i, x) \\ & \wedge\, d(c_j, c_t) > d(c_j, x) - d(c_i, x)\}\end{aligned} \tag{2}$$

Nevertheless, the noisy input vector still reduces the performance of this novel TIE approach.

Other studies [4], [5] used a multiple TIE (MTIE) approach to improve the TIE method by reducing the search space, increasing computation savings. But the selection rule for multiple dimensions of MTIE is still indefinite. The noisy input vector also reduces the performance of the novel MTIE approach.

A previous work [6] proposed a quasi-binary search (QBS) algorithm to improve computation of the VQ algorithm. This performance was better than TIE-1, TIE-2, and MTIE. The QBS algorithm improves on the TIE approach particularly for the noisy input signal. However, the quantization accuracy of the QBS algorithm is imperfect at 99.86%. The quantization accuracy of full-search and TIE approaches is 100%.

The TIE algorithm depends completely on the continuous and invariant input vectors. The potential for computation savings is great, but there are three drawbacks. First, computation savings in the image 'Baboon' are insufficient because the image is disordered. Second, in the VQ-based G.729 standard LSP encoder, a moving average (MA) filters the LSP parameter beforehand. The MA filter destroys the continuous property of the LSP bias. Thus, the performance of the TIE algorithm also decreases in the G.729 standard. Third, in the multi-stage G.729 standard VQ, there is a small codebook size and each codeword is neighborhood. Therefore, the TIE algorithms cannot work efficiently.

This study focuses on improving the TIE-based VQ approach. It proposes the RI-TIE algorithm and focuses on the noisy input signal. The image 'Baboon' and an LSP of G.729 verify the performance of the proposed method.

2. RI Rules and the TIE Algorithm

This study uses RI rules to improve the performance of the TIE algorithm. Figure 1 shows the TIE search table and the RI rule-based TIE in detail. For example, to complete the TIE process, select codeword C_3, which is the previous optimal codeword, as a reference codeword to find the nearest codeword group (NCG). Then, distance $2 \cdot d(X, C_3)$ between input vector X and reference codeword C_3 locates the NCG as candidate codewords. There are 51 candidate codewords in the first NCG. This reduces the search space from 128 to 51. This is the first search. To begin the second search, select the first candidate codeword C_{24} within the first NCG as the second reference codeword. Then, distance $2 \cdot d(X, C_{24})$ between input vector X and codeword C_{24} generates the second NCG. There are 40 candidate codewords in the second

Manuscript received April 12, 2012.
Manuscript revised August 10, 2012.
[†]The authors are with the Department of Electrical Engineering, National Taipei University of Technology, Taipei 10608, Taiwan.
[††]The author is with the Department of Electrical Engineering, National Chin-Yi University of Technology, Taichung 41170, Taiwan.
a) E-mail: hsf@ntut.edu.tw
DOI: 10.1587/transinf.E96.D.151

NCG. The search space decreases from 51 to 40. This is the second search. The number of times the search must be completed is now 51 to 40 + 2. The additional number represents the search of the first and second NCGs. For the third search, select the first candidate codeword C_{127} from the second NCG as the third reference codeword. The third NCG is generated and the search space is reduced from 40 to 8. The number of times the search must be completed is now (40 + 2) to (8 + 3). The recursive TIE clearly reduces the search space of the general TIE. If 11 searches are completed, an optimal codeword is produced. The number of searches is drastically reduced from 128 to 11.

In other words, if we select C_{24} within the 1st NCG and a number of NCG which is the input X and the codeword C_{29} to acquire the nearest codeword group as the candidate codewords are not less than 50 (50 − 1). In this situation, the second NCG cannot be generated; therefore, the second candidate codeword C_{29} in the first NCG is selected. Distance $2 \cdot d(X, C_{29})$ is placed between input X and codeword C_{29} to acquire the nearest codeword group as candidate codewords. The number of candidate codewords is less than 49 (51 − 2). The second candidate NCG is generated.

A number of new NCG are less than the number of original NCG minus the number of previous searches; thus, the new candidate NCG is generated. Repeat this process to the end.

Applying the intersection rule further reduces the TIE search space. In Fig. 1, the first time to acquire the nearest codeword group is the same as the general TIE [1]. There are 51 candidate codewords in the NCG. Applying the intersection rule reduces the search space from 51 to (8 + 3). The following are all best-matched codewords: the first codeword of the first NCG that has 51 candidate codewords is C_3, the first codeword of the second NCG that has 40 candidate codewords is C_{24}, and the first codeword of the third NCG that has 8 candidate codewords is C_{127}. Therefore, the intersection of these three NCGs is the best search space and represents codewords C_{29}, C_{44}, C_{98}, and C_{22}. The search space further reduces from (8 + 3) to (4 + 3). Thus, in Fig. 1, the search space is 51 for the general TIE and is (4 + 3) for the RI-TIE approach. The RI-TIE algorithm is listed as follows:

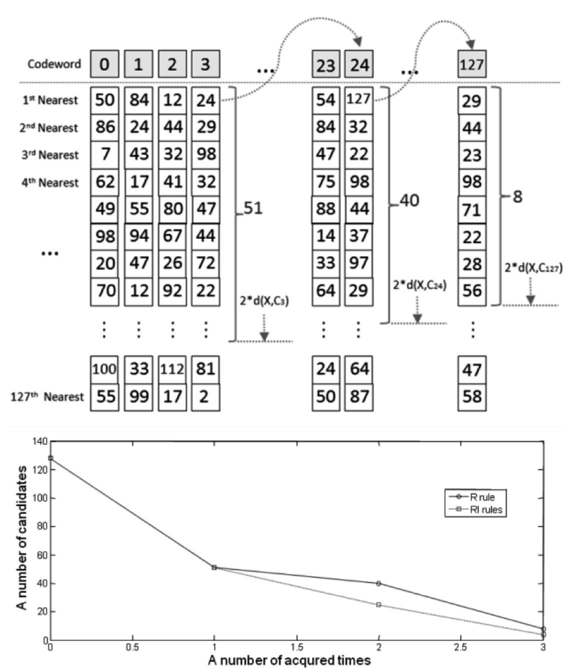

Fig. 1 The architecture and flowchart of the RI-TIE algorithm.

Step-1: Initialize TIE table, obtain input vector $\boldsymbol{x}$ and reference codeword $\boldsymbol{C}_j$.
Candidate group:

$$NCG(c_j) = \{c_t \,|\, d(c_j, c_t) < 2d(c_j, x)\}.$$

Codeword number: $NCG(c_j)$: $N(c_j)$.
Step-2: Repeat each codeword C_k from $NCG(c_j)$, $k = 1, 2, \ldots, N(c_j)$, as reference codeword.

$$\begin{aligned} &if(N(c_k) < N(c_j) - k), \quad NCG(c_j) = \\ &\quad NCG(c_j) \cap NCG(c_k), \quad k = 1. \end{aligned}$$

Otherwise continue until $k = N(c_j)$.

3. Experiment Results

This study used the VQ-based G.729 standard LSP encoder to examine the performance of the RI-TIE algorithm. Each speech group database for testing contained more than 300 seconds. Images with $2 \cdot 2$ pixels and 1024 codewords were also used to test the RI-TIE approach. Table 1 lists the computation savings of the TIE-1, TIE-2, MTIE, QBS, and RI-TIE approaches. The RI-TIE approach is superior to other approaches. Figure 2 shows that provided RI rules are made a few times in each test image, the smallest NCG is produced. On average, if RI rules are made six times, the smallest NCG is produced.

For the speech tests, the additional computation savings of the RI-TIE approach are 26.45% (= 62.71% −

Table 1 The computation saving rate of a G.729 LSP encoder and images encoder with TIE-1, TIE-2, MTIE, QBS, and RI-TIE algorithms.

Computation Savings		TIE-1	TIE-2	MTIE	QBS	RI-TIE	Contribution of RI Rules
Speech (LSP of G.729)	Voice-(Male)	36.26%	48.56%	46.88%	58.75%	62.71%	26.45%
	Voice-(Female)	42.97%	51.56%	53.91%	58.43%	72.66%	29.69%
	Music	29.69%	42.19%	42.97%	58.63%	55.47%	25.78%
	Noise	18.75%	28.91%	34.38%	59.05%	53.91%	35.16%
	Silence	94.53%	96.09%	94.53%	58.86%	96.09%	1.56%
Image (Pixel Size: 2·2) (codebook Size: 1024)	Baboon	67.73%	78.51%	73.48%	93.10%	94.65%	26.92%
	Barbara	76.04%	84.12%	82.68%	93.12%	94.71%	18.67%
	House	82.98%	88.90%	85.37%	93.24%	95.20%	12.22%
	Monarch	87.81%	91.80%	88.51%	93.15%	94.99%	7.18%
	Boats	87.84%	92.33%	89.53%	93.19%	96.97%	9.13%
	Pills	88.35%	92.72%	88.77%	93.20%	96.88%	8.53%
	Lena	90.22%	93.70%	91.46%	93.25%	98.08%	7.86%
	Jet-F16	90.93%	94.04%	92.14%	93.11%	98.21%	7.28%
	Pepper	91.99%	94.71%	92.53%	93.16%	97.73%	5.74%
	Tiffany	93.15%	95.56%	94.33%	93.15%	97.88%	4.73%
	Tomato	93.91%	96.25%	94.20%	93.17%	97.69%	3.78%

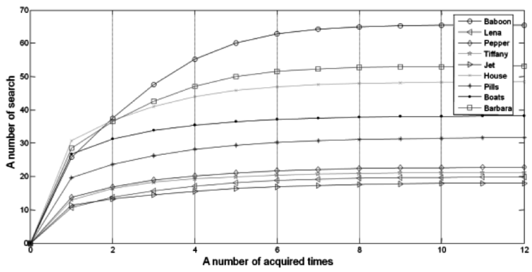

Fig. 2 Search times of the RI-TIE algorithm on image data.

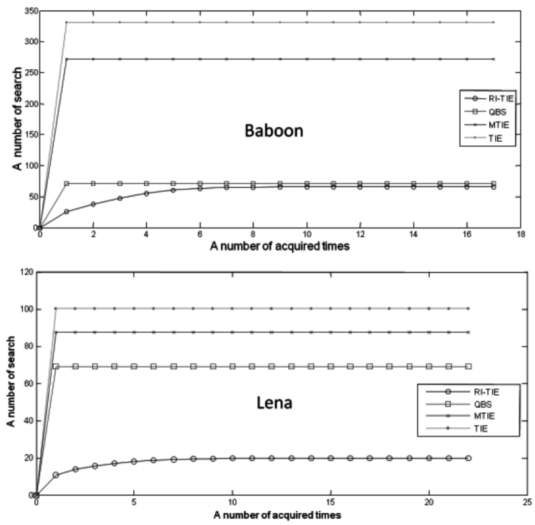

Fig. 3 Variety of image compression methods in ‘Baboon’ and ‘Lena’.

36.26%) and 29.69% (= 72.66% − 42.97%) for males and females, respectively. The additional computation savings of the RI-TIE approach are equal to the contribution of RI rules. The contribution of RI rules is defined as $\boldsymbol{C(RI)}$ and is listed as below:

$$\boldsymbol{C(RI) = CS(RI\text{-}TIE) - CS(TIE\text{-}1)} \quad (3)$$

The $\boldsymbol{CS(RI\text{-}TIE)}$ and $\boldsymbol{CS(TIE\text{-}1)}$ are the computation saving of RI-TIE and TIE-1 approaches.

For the image tests, the RI-TIE approach is independent of the continuous and invariant input signal and is superior to the TIE approach. The computation savings of the RI-TIE approach are better than those for the QBS approach. The additional computation savings of the RI-TIE approach are 26.92% (= 94.65% − 67.73%) (‘Baboon’) and 7.86% (= 98.08% − 90.22%) (‘Lena’).

Compared to the TIE algorithm, the RI-TIE approach is clearly more efficient on the noisy image. Figure 3 shows good results from either the smooth image ‘Lena’ or the noisy image ‘Baboon’. Experiment results confirm the good performance of the RI-TIE approach. The RI-TIE approach improves the weaknesses of the TIE method and outperforms the QBS approach. These results further confirm the compensation capacity of RI rules on the noisy input vector.

TIE-1, TIE-2, MTIE, and RI-TIE, use the same search theory as Full Search. The theory compares distance

Table 2 The PESQ of G.729 encoded speech with Full Search, TIE, RI-TIE, QBS, and Tree Search algorithms.

PESQ	Original Speech	G.729 Encoded Speech			
		Full Search/TIE	RI-TIE	QBS	Tree Search
Music	4.5	3.189	3.189	3.187	2.745
Noise	4.5	3.104	3.104	3.092	2.706
Female	4.5	3.494	3.494	3.487	3.084
Male	4.5	3.502	3.502	3.499	3.101

Table 3 The image PSNR with Full Search, TIE, RI-TIE, QBS, and Tree Search algorithms.

PSNR(dB)	Full Search/TIE	RI-TIE	QBS	Tree Search
Lena	34.184	34.184	34.181	25.283
Baboon	26.623	26.623	26.618	20.014
Jet	25.667	25.667	25.660	21.296
Pepper	27.205	27.205	27.198	20.228
Boats	28.645	28.645	28.641	21.571
House	32.376	32.376	32.371	24.744
Gold hill	22.733	22.733	22.728	14.332

individually. This method finds the optimum index, and optimum candidate indices are the same. Table 2 shows that the PESQ of each encoded speech actually is not different. Table 3 shows that the PSNR of each image is the same. It means that the distortion for TIE-1, TIE-2, MTIE, and RI-TIE and the distortion for Full search is the same. However, QBS and Tree Search approaches are different. It does not always produce the best index, and, in this case, there is distortion. Therefore, the PSNR of each image and the PESQ of each speech are the worst in the Tree Search approach. The TIE approach listed in Tables 2 and 3 includes TIE-1, TIE-2, and MTIE approaches.

For the Full Search, the image search is 1024 and the speech search is 128. The last best index and input vector determine the search spaces for TIE-1, TIE-2, and MTIE. Thus, their search space is fixed. The RI-TIE method selects the last best index to acquire the nearest codeword group. After each search, the search space gradually decreases.

RI-TIE finds the optimal index faster than TIE-1, TIE-2, and MTIE. We require the lowest search times. Therefore, the optimal solution saves time in computation.

These results confirm the outstanding performance of the RI-TIE approach. The RI-TIE approach solves the weaknesses of the TIE method, and outperforms the Tree Search approach without losing quantization accuracy. The performance of RI-TIE approach is also superior to QBS approach also.

The RI-TIE approach is independent of the continuous and invariant input signal and is superior to above approach. It has good results from either the smooth image ‘Lena’ or the noisy image ‘Baboon’. Of course, the speech of female or male is also very effective.

4. Conclusion

This study presents the RI-TIE approach for improving the performance of the TIE algorithm. The VQ-based G.729 standard LSP encoder and the noisy image ‘Baboon’ were used to examine the effects of the RI rules. Experiment results show that additional computation savings of the RI-TIE approach are 29.69% (= 62.71% − 36.26%) and 26.92% (= 94.65% − 67.73%) for the female speech and the image ‘Baboon,’ respectively. Results also show that the RI-TIE approach performs better than the QBS approach. The RI rules compensate for the weakness of TIE algorithms. This is a valuable method for compensating and improving the deficiencies of TIE-based VQ encoding.

References

[1] S.H. Hwang and S.H. Chen, “Fast encoding algorithm for VQ-based image coding,” Electron. Lett., vol.26, no.19, pp.1618–1619, 1990.
[2] M. Kamel and L. Guan, “Fast nearest neighbor search for vector quantization of image data,” Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), vol.3, pp.623–626, Aug. 1992.
[3] W. Li and E.A. Salari, “Fast vector quantization encoding method for image compression,” IEEE Trans. Circuits Syst. Video Technol., vol.15, no.2, pp.119–123, 1995.
[4] S.Y. Choi and S.I. Chae, “Incremental-search fast vector quantiser using triangular inequalities for multiple anchors,” Electron. Lett., vol.34, no.12, pp.1192–1193, 1998.
[5] C.H. Hsieh and Y.J. Liu, “Fast search algorithms for vector quantization of images using multiple triangle inequalities and wavelet transform,” IEEE Trans. Image Process., vol.19, no.30, pp.321–328, 2000.
[6] L.J. Yan and S.H. Hwang, “Vector quantisation base on quasi-search algorithm,” IET Image Processing, vol.5, no.1, pp.49–54, 2011.