A NEW PARTIAL CODEWORD UPDATING SCHEME BASED ON RATE-DISTORTION OPTIMIZATION FOR ADAPTIVE VECTOR QUANTIZATION

Kai Guo and Lai-Man Po

Department of Electronic Engineering, City University of Hong Kong
Tat Chi Avenue, Kowloon, Hong Kong, China

ABSTRACT

In this paper, we propose a new adaptive vector quantization (AVQ) algorithm based on the rate-distortion optimization. This algorithm employs a new partial codeword updating (PCU) scheme which achieves rate-distortion performance superior to that of the conventional AVQ algorithms using the full codeword updating (FCU) scheme. The PCU-AVQ only updates the codeword's components with the quantization error higher than an optimal threshold instead of replacing the whole codeword. Additionally, the mathematical relation between the Lagrangian multiplier and the approximate optimal threshold is devised to reduce the rate-distortion cost computation. The experimental results show that the proposed PCU-AVQ algorithm indeed improves the rate-distortion performance without much computational complexity penalty. The PCU-AVQ can be combined with transform coding and entropy coding for higher compression ratio, and it can be widely implemented in specific AVQ algorithms for image, video and speech coding.

Index Terms**—** Adaptive vector quantization, image and video coding, rate-distortion optimization

1. INTRODUCTION

Vector quantization (VQ) is a powerful data compression technique successfully employed in various applications[1]. A great advantage of VQ is that the decoding process is quite simple; therefore, VQ is suitable for the single encoder, multiple decoder systems. VQ is theoretically attractive; however, there still exists a large gap between the theoretical performance and actually achieved performance. Thus, many adaptive vector quantization (AVQ) algorithms[2],[3],[4] were proposed. The most important feature of AVQ is that codebook can be updated to track the changing statistics of data source in the coding process[2].

Many AVQ algorithms focus on the minimization of distortion alone. As a result, the overhead of bit rate may worsen the rate-distortion performance even if the distortion is quite small. In order to reach an optimized tradeoff between the distortion and the rate, a criterion involving both rate and distortion should be used[5]. The previous AVQ algorithms update one codeword of the codebook at a time, which is called ”full codeword updating (FCU)” scheme. In this paper, we propose a new AVQ algorithm with the partial codeword updating (PCU) scheme. This approach partially updates a codeword and can improve the rate-distortion performance.

AVQ algorithms are not always used alone, i.e., they are always along with a transform coding stage prior to AVQ. Moreover, entropy coding is also employed in AVQ to further increase the compression ratio. It is believed that our proposed AVQ algorithm can also combine with transform coding as well as entropy coding and lead to better performance. In addition, the proposed AVQ algorithm can adapt to diverse data sources, such as image, video and speech signals.

The organization of this paper is that the conventional FCU-AVQ algorithm is presented in Section 2. In Section 3, the proposed PCU-AVQ algorithm is presented. In Section 4, the relationship between Lagrangian multiplier and approximate optimal threshold is built. Simulation results and conclusion are presented in Section 5 and 6, respectively.

2. ADAPTIVE VECTOR QUANTIZATION BASED ON RATE-DISTORTION CRITERION

An adaptive vector quantization process can be described as[3] $Q_t : \Re^K \rightarrow C_t$, where Q_t is a time-varying mapping of vector in K-dimensional Euclidean space $\Re^K$ into the local codebook C_t. The output of an AVQ system consists of indices and side information, which indicates if a codebook updating happens and the updating information when a codebook is updated. In order to achieve an optimized tradeoff between the distortion and rate, we introduce a rate-distortion cost[5] $J(\mathbf{x}, \mathbf{y}; \lambda) = d(\mathbf{x}, \mathbf{y}) + \lambda \cdot r(\mathbf{y})$, where $\mathbf{x}, \mathbf{y}$ are two vectors, $d(\mathbf{x}, \mathbf{y})$ and $r(\mathbf{y})$ represent the distortion and bit rate. The Lagrangian multiplier λ is used to adjust the tradeoff between the rate and distortion.

The basic encoding process of conventional R-D cost based FCU-AVQ algorithm is summarized as below:
1. Initialize the codebook, $\mathbf{P}$, with size $N = |\mathbf{P}|$, and specify the Lagrangian multiplier λ.

The work was supported by the grant from City University of Hong Kong [Project No.*7001853*]

2. Find the nearest codeword $\hat{\mathbf{p}}$ for the input vector $\mathbf{s}$ based on the Euclidean norm: $\|\mathbf{s} - \hat{\mathbf{p}}\|_2 = \min \|\mathbf{s} - \mathbf{p_i}\|_2$

3. Compute the R-D cost if no updating happens:
$J_1 = d(\mathbf{s}, \hat{\mathbf{p}}) + \lambda \cdot r(i)$, where $r(i)$ is the bit rate for index

4. Compute the R-D cost if the codebook updating happens:
$J_2 = \lambda[r(i) + r(\mathbf{s})]$, where $r(\mathbf{s})$ is the bit rate consumed for the codebook updating.

5.If $J_1 < J_2$, no updating happens; else, $\hat{\mathbf{p}}$ is replaced by $\mathbf{s}$.

6. Entropy codes the index i, then transmit index and side information to the decoder.

In the encoding process, R-D cost is used to determine whether the codebook needs update or not. In practical applications, the AVQ is usually applied in the transform domain and entropy coding is also used to encode the index i and/or the side information in step 6 for achieving higher performance. However, transform coding and entropy coding are not considered in this paper as we mainly focus on the rate-distortion performance improvement of the AVQ using the new PCU scheme.

3. AVQ BASED ON THE PARTIAL CODEWORD UPDATING SCHEME

The FCU-AVQ algorithms update the whole codeword at a time which leads to zero distortion for the input vector[2]. In many cases, however, it is a waste of bits to replace the whole codeword with the input vector. So better rate distortion performance can be achieved if we only update those codeword components whose quantization error exceeds a threshold.

3.1. Description of Partial Codeword Updating Scheme

The proposed PCU-AVQ algorithm uses integer thresholds to update the nearest codeword's components with large quantization error. The optimal threshold $\hat{T}$ is selected from $\Phi = \{T_k = k - 1, k = 1, 2, \cdots, M - 1\}$, where T_k presents the possible threshold and M is the maximum quantization level. For example, in image VQ with 8-bit per pixel, then T_k is ranged from 0 to 255 with $M = 256$. To find the optimal threshold value, full search technique is employed based on the rate-distortion criterion. The PCU-AVQ is described as follows:

1: Initialize the codebook $\mathbf{P}$, and Lagrangian multiplier λ.

2: Find the nearest codeword $\hat{\mathbf{p}}$ for the input vector s based on the Euclidean norm: $\|\mathbf{s} - \hat{\mathbf{p}}\|_2 = \min \|\mathbf{s} - \mathbf{p_i}\|_2$

3: Specify a threshold T_k from Φ. Generate the partial updated codeword $\tilde{\mathbf{p}}(T_k)$ and record the updated components location into vector $\mathbf{u}(T_k)$, based on the absolute difference between the components of the input vector and the nearest codeword by comparing the difference with the threshold T.

$$\tilde{p_j}(T_k) = \begin{cases} s_j, & \text{if } |s_j - \hat{p_j}| > T_k \\ \hat{p_j}, & \text{otherwise} \end{cases}$$

$$\tilde{u}_j(T_k) = \begin{cases} 1, & \text{if } |s_j - \hat{p_j}| > T_k \\ 0, & \text{otherwise} \end{cases}$$

4: Compute the PCU-based R-D cost for threshold T_k:
$J_3(T_k) = d(\mathbf{s}, \tilde{\mathbf{p}}(T_k)) + \lambda[r(i) + r(\tilde{\mathbf{p}}(T_k)) + r(\mathbf{u}(T_k))]$

5: Repeat Step 3 and Step 4 to calculate the R-D cost for each threshold value:$J_3(T_1), J_3(T_2), \cdots, J_3(T_M)$. Then determine the optimal threshold $\hat{T}$ that leads to the minimum R-D cost:$J_{3min} = J_3(\hat{T}) = \min J_3(T_1), J_3(T_2), \cdots, J_3(T_M)$

6: Compute the R-D cost: $J_1 = d(\mathbf{s}, \hat{\mathbf{p}}) + \lambda r(i)$ (non-updating mode), $J_2 = \lambda[r(i) + r(\mathbf{s})]$ (FCU mode)
If $J_1 = min(J_1, J_2 J_{3min})$, no updating happens;
If $J_2 = min(J_1, J_2 J_{3min})$, FCU mode is selected;
If $J_{3min} = min(J_1, J_2 J_{3min})$, PCU mode is selected;

Note that the proposed AVQ algorithm is based on rate-distortion criterion in two aspects. Firstly, the rate-distortion measure is used to select the optimal threshold $\hat{T}$ from Φ. Secondly, the rate-distortion measure selects the best mode among non-updated mode, FCU mode and PCU mode. Thus, the PCU-AVQ algorithm is possible to achieve better rate-distortion performance than the FCU-AVQ algorithm.

3.2. Side Information Encoding Method

In the proposed PCU-AVQ algorithm, a vector $\mathbf{u}$ is needed to represent which components are required to update. Normally, for a *K*-dimension codeword, a *K*-length binary sequence is needed to record the updated components. For example, in the sequence, 0001001001000110000, "0" indicates the positions without updating and "1" indicates the positions where updating happens. In some cases, when the number of updated components is quite small ("1" is seldom) or quite large ("1" is frequent), we can directly record the positions of "1" or the positions of "0" so that more bits can be saved.

In this way, fewer bits are used to present the side information and the mode classification method is given below:

1. Count the number of positions where updating happens: *n*

2. Calculate the nearest integer of $\frac{K}{\log_2 K}$ towards the zero direction: $[\frac{K}{\log_2 K}]$

3. If $n < [\frac{K}{\log_2 K}]$ or $n > K - [\frac{K}{\log_2 K}]$, record each positions of "1" or "0" with $\log_2 K$-bit numbers;
Else if $[\frac{K}{\log_2 K}] \leq n \leq K - [\frac{K}{\log_2 K}]$, present the side information with *K*-length binary sequence.

4. RELATIONSHIP BETWEEN UPDATING THRESHOLD AND LAGRANGIAN MULTIPLIER

In the PCU-AVQ algorithm as proposed in the last section, we use the full search technique to find the optimal threshold $\hat{T}$, which is very computational intensive and will limit the practical implementation of the proposed algorithm. Therefore, an efficient way to find the optimal threshold is necessary. $\hat{T}$ is actually related to the Lagrangian multiplier λ, which controls the tradeoff between rate and distortion[5]. A small

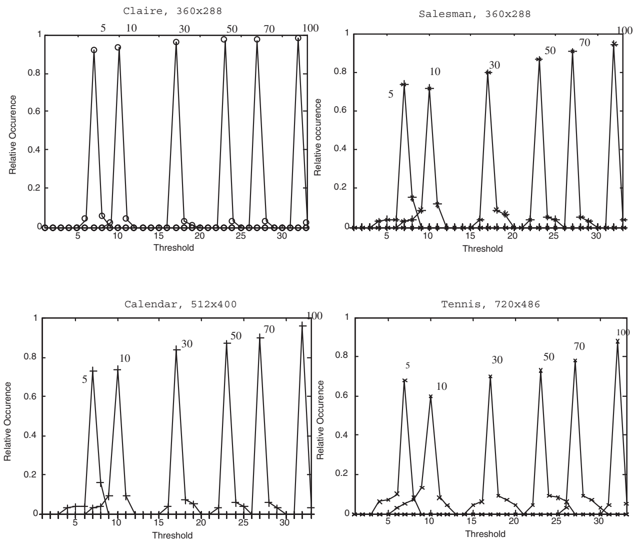

Fig. 1. Relative occurrences of optimal thresholds

Table 1. Relation between T^* and λ

Threshold T^*	7	10	17	23	27	32
Lagrangian Multiplier λ	5	10	30	50	70	100

λ emphasizes distortion, so $\hat{T}$ should be smaller in order to reduce the distortion. A large λ emphasizes bit rate, so $\hat{T}$ should be larger in order to control the bit rate. In H.264 coding standard, R-D cost is also used for mode decision and the Lagrangian multiplier λ is related to the quantization parameter (QP)[6]. In addition, a strong connection between λ and QP is experimentally built, by fixing λ and finding the optimal QP that minimizes R-D cost. $\hat{T}$ in the PCU-AVQ is similar to the QP in H.264; thus, the similar method can be used to find the relationship between $\hat{T}$ and λ. The Lagrangian multiplier λ is varied over six values: 5, 10, 30, 50, 70 and 100, producing six histograms which show the occurrence frequency of optimal $\hat{T}$ in Fig. 1. Based on the results in Fig. 1, we can observe that:

1. For a specified λ, there always exists an appropriate threshold T^* whose probability to be the optimal threshold $\hat{T}$ is much higher than other thresholds;

2. The value of the threshold T^* is nearly not varied with different video sequences.

The first observation provides the feasibility to determine a good threshold in advance without using the full search technique. Since the threshold T^* is much more likely to be the optimal threshold $\hat{T}$ than other thresholds, it can be considered as an efficient estimation of $\hat{T}$. The second observation indicates the robust relationship between T^* and λ; thus, their relationship is widely applicable. As long as we can build a mathematical relation between T^* and λ, it is convenient to determine a good threshold for a specified λ.

A typical approximation relation between distortion and rate is [5]: $R(D) = a\ln(\frac{\sigma^2}{D})$, where a and σ^2 are constant. Then, $J = D + \lambda \cdot R = D + \lambda \cdot a\ln(\frac{\sigma^2}{D})$.

The minimization of *J* for a given λ can be reached when the derivative of *J* with respect to *D* is equal to zero.

$$\frac{dJ}{dD} = 1 - \frac{a\lambda}{D} = 0 \Rightarrow \lambda = \frac{D}{a} \tag{1}$$

We define the component quantization error as: $e_j = |s_j - p_{ij}|$. After PCU scheme, definitely, the component's quantization error is limited within the range of the threshold. We define components that are not updated as $e_1^{'}, e_2^{'}, \ldots, e_{K-n}^{'}$. *n* is the number of updated components. Besides, a reasonable probability distribution can be approximated as a constant within the threshold interval at a sufficient bit rate: $p(e_j^{'}) = \frac{1}{\hat{T}}, e_j^{'} \in [0, \hat{T}]$. In this paper, the distortion *D* uses the Euclidean norm $D = \sum_{j=1}^{K-n} e_j^{'2}$. The mean of D is used to estimate D:

$$\begin{aligned} E(D) &= (K-n)E(e_j^{'2}) \\ &= (K-n)\int_0^{\hat{T}} e_j^{'2} p(e_j^{'}) de_j^{'} = \frac{(K-n)\hat{T}}{3} \end{aligned} \tag{2}$$

Insert(2)into(1):

$$\lambda = \frac{D}{a} \cong \frac{K-n}{3a}(\hat{T})^2 = \mu(\hat{T})^2 \tag{3}$$

Although (3) is based on the mean of D, it indicates that it may be reasonable for the Lagrangian multiplier λ to be proportional to the square of the optimal threshold $\hat{T}$. In the coding process, $\hat{T}$ is not fixed and in most cases $\hat{T} = T^*$; therefore, the coefficient μ is also varied. In order to build the relationship between λ and $\hat{T}$, we use $\bar{\mu}$ and T^* to estimate μ and $\hat{T}$, respectively. Table 1 shows the experimental connection between T^* and λ. We find that $\bar{\mu} = 0.10$ is able to fit the data in Table 1 very well. Fig. 2 compares the experimental curve and fitting curve when $\bar{\mu} = 0.10$ and it indicates that the fitting error is quite small, which also supports the assumption in (3). Based on the discussion above, we can obtain an experience equation:

$$\mathbf{T}^* = \sqrt{\frac{\lambda}{0.10}} \tag{4}$$

With the use of this equation, it is very convenient to determine the approximate optimal threshold T^* without full search technique. Therefore, the proposed PCU-AVQ algorithm can be simplified as follows:

1:Initialize the codebook $\mathbf{P}$, and Lagrangian multiplier λ. Calculate T^* according to (4): $\mathbf{T}^* = \sqrt{\lambda/0.10}$

2. Find the nearest codeword $\hat{\mathbf{p}}$ for the input vector $\mathbf{s}$ based on the Euclidean norm: $\|\mathbf{s} - \hat{\mathbf{p}}\|_2 = \min \|\mathbf{s} - \mathbf{p_i}\|_2$

3: Generate the partial updated codeword $\tilde{\mathbf{p}}(T^*)$ and record

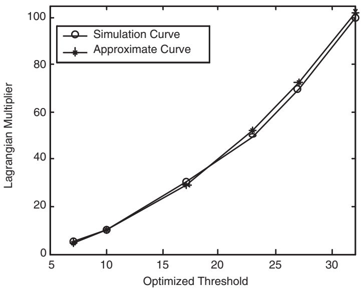

Fig. 2. Comparison between the experimental and approximate relation between T^* and λ

the updated components location into vector $\mathbf{u}(T^*)$.

$$\tilde{p}_j(T^*) = \begin{cases} s_j, & \text{if } |s_j - \hat{p_j}| > T^* \\ \hat{p}_j, & \text{otherwise} \end{cases}$$

$$\tilde{u}_j(T^*) = \begin{cases} 1, & \text{if } |s_j - \hat{p_j}| > T^* \\ 0, & \text{otherwise} \end{cases}$$

4: Compute R-D cost $J_3 = d(\mathbf{s}, \tilde{\mathbf{p}}(T^*)) + \lambda[r(i) + r(\tilde{\mathbf{p}}(T^*)) + r(\mathbf{u}(T^*))]$ (PCU mode), $J_1 = d(\mathbf{s}, \hat{\mathbf{p}}) + \lambda r(i)$ (non-updating mode), $J_2 = \lambda[r(i) + r(\mathbf{s})]$ (FCU mode)
If $J_1 = min(J_1, J_2 J_3)$, no updating happens;
If $J_2 = min(J_1, J_2 J_3)$, FCU mode is selected;
If $J_3 = min(J_1, J_2 J_3)$, PCU mode is selected;

5. SIMULATION RESULTS

The proposed PCU-AVQ algorithm based on partial codeword updating scheme was tested using the first 50 frames from four video sequences: Claire, Salesman, Calendar and Tennis. The FCU-AVQ for simulation is described in Section 2. Since this paper mainly contributes to the improvement of rate distortion performance, the simulation does not implement the transform coding and entropy coding. Although the simulation is based on video sequences, we expect to obtain a general conclusion for data sources. Thereby, no specific video coding technologies are utilized in both the PCU-AVQ and FCU-AVQ algorithms. From the simulation results, we can conclude that the proposed AVQ algorithm can improve PSNR by around 0.35 to 1.75 dB in different Lagrangian multiplier. The comparison of rate-distortion curves is shown in Fig. 3. It is believed that the proposed AVQ algorithm can achieve better rate-distortion performance if combined with transform coding and entropy coding techniques.

6. CONCLUSIONS

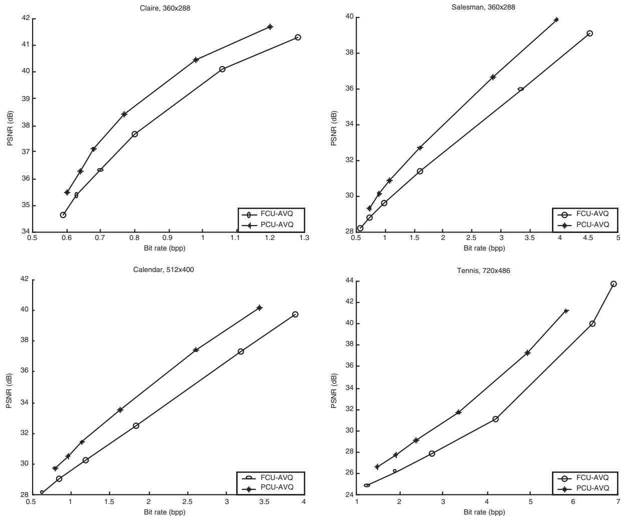

Fig. 3. Comparison of rate-distortion curves between the FCU-AVQ and the PCU-AVQ

In this paper, a new AVQ algorithm based on the partial codeword updating scheme is proposed, which efficiently improves the rate-distortion performance compared with conventional AVQ algorithms. In order to avoid the full search process for the optimal threshold, we build the relationship between the Lagrangian multiplier and the approximate optimal threshold. From the simulation results, the proposed PCU-AVQ algorithm can improve PSNR by around 0.35 to 1.75 dB, which shows the efficiency of the PCU-AVQ algorithm.

7. REFERENCES

[1] A. Gersho and R. M. Gray, ”Vector Quantization and Signal Compression”, Boston, MA: Kluwer, 1992

[2] G. Shen, B. Zeng and Liou, M. -L., ”Adaptive vector quantization with codebook updating based on locality and history,” *IEEE Trans. on Image Processing*, Vol. 12, pp.283-295, Mar. 2003

[3] J. E. Fowler, ”Generalized threshold replenishment: an adaptive vector quantization algorithm for the coding of nonstationary sources”, *IEEE Trans. on Image Processing*, Vol. 7, pp.1410-1424, Oct. 1998

[4] W. B. Mikhael and P. Ragothaman, ”Adaptive vector quantization of non-orthogonal representations for image compression”, *IEEE Electronics Letters*, Vol. 39, pp. 200-201, Jan.2003

[5] G. J. Sullivan and T. Wiegand, ”Rate-distortion optimization for video compression”, *IEEE Signal Processing Magazine*, Vol. 15, pp.74-90, Nov.1998.

[6] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, ” Overview of the H.264/AVC video coding standard,” *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 13, pp.560-576, July 2003