ABSTRACT

In this work we study the performance of Tree-Structured Vector Quantizers (TSVQ) jointly designed with Forward Error Correction Code providing Unequal Error Protection (UEP) for the transmission of still images over noisy channels. Comparisons have shown that this scheme performs better than Channel Optimized Vector Quantizers (COVQ) scheme and is less affected by the channel distortion. The design of the TSVQ matched to the noisy channel is less complex than the design of COVQ and it has, in addition, a simpler implementation.

1. INTRODUCTION

To transmit data over a noisy channel requires, in most applications, some kind of error protection. This is the case for instance in the transmission of images captured by remote sensing satellites (RSS) which is the scenario that motivates the present work. Usually the introduction of error control increases the bandwidth and when the latter is limited, data compression is needed to compensate for the additional bit rate introduced by the channel coding. Shannon’s well-known separation principle states that one can design and code the source and channel separately with an optimal result provided the compressed and coded block-length are large enough and, the channel is time-invariant. This however is an asymptotic result and, in real-world applications this might not be true since neither is the channel constant with respect to time nor is the block-length large enough for the principle to hold. In this case, joint source/channel coding schemes have proved to be an alternative with significant gains in performance. This immediately leads to the design of joint source-channel quantizers.

Several approaches to vector quantization (VQ) under channel constraint have been proposed in the literature. Source-optimized VQs by efficient index assignment can lead to good results over noisy channels. Algorithms for efficient index assignment are reported in [10, 6, 1]. Source-optimized scheme reported in [9], where the bit-stream of the SPIHT coder is packetized into fixed-length blocks and protected with RCPC codes and error-detection codes, renders astounding performance in SNR yet it has a non-negligible probability of incomplete image decoding.

Another approach is the COVQ proposed in [1, 7], which basically consists in an algorithm that, by taking into account the channel crossover probabilities, produces a codebook optimized for this channel. The resulting quantizer has a better performance when compared to the plain VQ operating over noisy environment since it trades off quantizer accuracy for channel-error resilience. The performance of the COVQ scheme is locally optimal but yet the channel noise introduces an annoying distortion.

The COVQ design can be further improved channel by use of so-called channel noise relaxation [2]. Starting with a channel with a larger error probability, the quantizer can be better designed by relaxing the channel error probability till one gets to the desired channel. This therefore prevents the design from converging to a poor local optimum.

The straightforward solution to the transmission of quantized data over a noisy channel is tandem source-channel coding scheme. Data is quantized with a VQ designed for a noiseless channel [3] and a channel code is used to reduce or even eliminate the effects of the channel noise. This results in a scheme that is not affected by the channel noise but the bits wasted with the code results in large quantization distortion.

Another scheme which jointly design the source and channel codes is reported in [4]. By combining the COVQ with RCPC codes, an iterative design is obtained with a significant gain over COVQ. This gain is mainly due to soft decision inherent to the RCPC code and also due to an optimized bit allocation between channel and source codes. A disadvantage of such scheme however is that it does not allow for UEP since there is no order of importance in the index bits.

In [8], an algorithm to design TSVQ matched to the channel was introduced. This is an efficient design technique that although not producing an optimal quantizer has the advan-
tage of lowering the computational complexity (of the design and the encoding). In addition, the successive refining characteristic of the TSVQ make it amenable to be used with FEC. In this work we study the performance of TSVQ used with UEP over noisy channels which is an intermediate scheme between the COVQ (which does not explicitly uses error correction) and the VQ + FEC tandem scheme. The paper is organized as follows: the formulation of the “Tree-Structured Vector Quantizer” matched to noisy channel is presented in Section 2 with an efficient algorithm to design the quantizer. The problem of transmission of quantized images through noisy channels is discussed on Section 3. Results are presented on Section 4 with concluding remarks on Section 5.

2. TREE STRUCTURED VECTOR QUANTIZATION OF IMAGES FOR NOISY CHANNELS

The Tree-structured Vector Quantization of an image $I$ can be understood as starting with a vectorization that maps consecutive $L = w \times b$ image cells into a sequence $\{x_n\} = (x_1, \ldots, x_n, \ldots, x_N)$ of $L$-dimensional vectors. This sequence is then submitted to a succession of $\Lambda$ quantizers (we will restrict ourselves to binary codebook quantizers which output an index 0 or 1 if the input vector is closer to centroid 0 or 1). All the indices $\{b_{\lambda,n}\}$ output by the quantizer on layer $\lambda$ ($\lambda = 1, \ldots, \Lambda$) are sent through a Binary Symmetric Channel (BSC). The codebook used by TSVQ on the first layer is $C_1 = \{y_1^0, y_1^1\}$. The codebooks used by TSVQ on layer $\lambda$ ($\lambda > 1$) are represented by $C_{\lambda}^{[b_{\lambda-1,n}]} = \{y_{\lambda-1,n}^0, y_{\lambda-1,n}^1\}$, where the block which determines this codebook, the length $\lambda - 1$ block of binary indices $b_{\lambda-1,n} = (b_{1,n}, \ldots, b_{\lambda-1,n})$ is defined as $S_t = 0$. The reconstructed cell delivered to the user is then the $L$-dimensional vector $\hat{x}_n = (y_{b_{1,n}, \ldots, b_{\lambda,n}})$.

Distortion between the vector $x_n$ output by the source and the corresponding reconstructed vector $\hat{x}_n$ delivered at the receiving end will be measured by the squared norm of the difference $d(x_n, \hat{x}_n) = \|x_n - \hat{x}_n\|^2$.

An optimum TSVQ for noiseless channel [8], can be obtained by finding a collection $\mathcal{F}$ of $\sum_{\lambda=0}^{\Lambda-1} 2^\lambda$ codebooks,

$$\mathcal{F} = \{C_0, C^{[b]}_{\lambda} : (\lambda, b_{\lambda}) \in \{1, \ldots, \Lambda\} \times \{0, 1\}^\lambda\},$$

that yields an overall minimum average quantization distortion. By using the LBG algorithm [3] one can design a locally optimal TSVQ — but its performance is poor on noisy channel since the channel is not taken into account by the design.

Finding an optimum TSVQ for the noisy channel, i.e. finding the collection $\mathcal{F}$ that minimizes the average distortion, is a much harder problem. For a real valued vector source with probability density function $p(x)$, an $M$-level TSVQ and a channel described by a transition probability $P(j|i)$ between the transmitted vector index $i$ and received index $j$, the average distortion can be written as

$$D(\mathcal{F}) = \frac{1}{L} \sum_{i=1}^{M} \sum_{j=1}^{M} P(j|i) d(i, j)$$

where $\{S_1, \ldots, S_M\}$ is the partition of the $L$-dimensional space induced by $\mathcal{F}$ and

$$d(i, j) = P(j|i) \int_{S_i} p(x) d(x, c_j) dx$$

is the average distortion when a vector $x \in S_i$ ends up mapped, by the TSVQ and BSC in the centroid $c_j$.

An algorithm is presented in [8] for designing TSVQ which takes into account the noisy channel. The procedure is called a CM-TSVQ-design since it does not guarantee an optimal encoder-decoder pair. We present an algorithm next which uses a training set instead of a known $p(x)$, developed along the same lines as the CM-TSVQ to design the quantizer.

2.1. Channel-Matched Tree Structured Vector Quantizer algorithm

CM-TSVQ Design Algorithm

Step 1 Let the training set be $S_0 = \{x_n\}_{n=1}^N$, and let the evaluated initial centroid be

$$y_0 = \frac{\sum_{n=0}^{N-1} x_n}{N}. \quad (3)$$

Step 2 $\lambda = 1$

Get the first layer codebook $C_1$ by creating two new centroids derived from the previous: initially set $y_1^0 = y_0^0$ and $y_1^1 = y_0^0 + \varepsilon$, (where $\varepsilon$ is a perturbation vector used to split the previous centroid in two) and then use the LBG algorithm to get the optimum codebook for the first layer quantizer. The partition $\{S_0^0, S_1^1\}$ of $S_0$ induced by $C_1$ are two new training set associated with the index 0 and 1, respectively.

Step 3 $\lambda = \lambda + 1$.

Find $C_{\lambda}^{[b_{\lambda-1},0]} = \{y_{\lambda-1,n}^{[b_{\lambda-1,0}]}, y_{\lambda-1,n}^{[b_{\lambda-1,1}]}\}$, the codebook associated to each length $\lambda-1$ distinct block of binary indices
The design (and coding) complexity of the CM-TSVQ is much lower than the complexity of COVQ even when a fast implementation of COVQ [5] is used.

3. TRANSMISSION OF QUANTIZED IMAGES THROUGH NOISY CHANNELS

Images VQ compressed and transmitted through noisy channels are delivered to the user with an overall distortion which is a combination of two terms: the quantization distortion and the channel error distortion. Channel optimized vector quantizers COVQ, designed by taking the channel error into account, render a performance better than plain VQ but are still perturbed by an annoying distortion since the transmitted index \( i \) (i.e., \( b_1, \ldots, b_{\Lambda,n} \)) is randomly transformed into an index \( j \) (i.e., \( b_1, \ldots, b_{\Lambda,n} \)). An way to attenuate this problem is to use Forward Error Correct Codes (FEC) — the channel bit error rate is reduced but at the expense of an increased distortion due to quantization. The burden on the rate put by VQ in tandem with FEC can be overcome by using CM-TSVQ and protecting with FEC only the TSVQ lower layer bits which causes larger damage to the transmission when delivered in error. This is equivalent to providing UEP (Unequal Error Protection) to the transmission.

The study of TSVQ with UEP is the subject of the present work. The diagram in Fig. 1 illustrates the general structure of such a TSVQ, with an indication that the binary data output by the TSVQ can be processed for protection before being sent to the channel. We have investigated two kinds of processing: (1) FEC codes and (2) FEC codes on reverse. The encoding process is done by entering successive blocks of bits of a given layer, say \( b_{\lambda, n+1} \), into the rate \( R_{\lambda} = \frac{1}{7} \) FEC encoder and transmitting the corresponding output \( u_{\lambda, n+1} \). FEC on reverse means that the TSVQ bit-stream is passed through a FEC decoder which introduces errors whenever the input block does not constitute a legitimate binary codeword. Notice that in this case \( \frac{1}{7} \) meaning no rate increase. For the layers protected with FEC, processing is meant to be FEC while for other layers it means “no processing”.

Several FEC schemes can be used with the TSVQ. We have chosen to make our investigation by using simple linear block codes. We have performed simulations considering transmission through a BSC with cross-over probability \( p_0 = .001 \), and used Hamming (7, 4) and (15, 11) codes. Allocation of code rates \( R_{\lambda} \) through the layers were done by trial and error — the lower layers bits were coded with either a rate \( \frac{1}{7} \) code or rate \( \frac{11}{15} \). One can easily verify that, with these codes, virtually error-free transmission can be attained — the probability of a binary block with two bits in error is about \( 10^{-5} \) and \( 10^{-3} \) for the Hamming (7, 4) and (15, 11) respectively. The simulations results are presented next.

4. RESULTS

Extensive simulation results have shown that a better performance, compared to COVQ, can be achieved when using the CM-TSVQ with UEP scheme. Fig. 2 displays the PSNR × Rate performance of the CM-TSVQ when compressing the image Lena (512 × 512). In this same figure the corresponding curves of the COVQ, VQ and VQ + FEC are also plotted exhibiting the superior performance of CM-TSVQ. This superior performance can be explained by the use of a form of channel relaxation in the quantizer design. We noted in our simulations that the performance could be improved by designing the first stages of the TSVQ with a larger probability of error (in our experiments we have used \( P_e \approx 0.2 \)).
It must be pointed out that although CM-TSVQ has a slightly better PSNR performance, it still exhibits typical COVQ artifacts (salt-and-pepper like noise). The use of UEP allows us to overcome this problem. The protection of the TSVQ lower layer bit stream with FEC reduces the impact of the channel noise. Tables 1 and 2 list the PSNR results obtained with some of the simulated schemes. On Table 1, the schemes are described by the following notation: scheme $H^{1-3}0^*$ works with each of the first three layers protected with an H(15,11) FEC code; scheme $H^{1+2}H0^*$ $H^{1+2}H^{3+4}0^*$ works with the bit-stream from the first and second layers interleaved and protected with an H(15,11), plus an H(15,11) on the third layer; scheme $H^{1+2,3+4,5+6,7}0^*$ works with an interleaved H(15,11) on layers 1+2, 3+4 and 5+6, and an H(15,11) code on the seventh layer; finally the scheme $H_0^*D^{1-4}$ works with an H(15,11) code on reverse on the first and second layers. $P_e = 0.2$ is the probability of error for the first TSVQ stage. The schemes on Table 2 are similarly described.

For natural images, the CM-TSVQ plus FEC has a near 1dB improvement in PSNR over COVQ. For RSS images such improvement was not observed. In both cases however, the decoded images are subjectively better as illustrated in Fig. 3 and Fig. 4 (salt-and-pepper like noise is drastically reduced). It should be pointed out that CM-TSVQ has a lower computational complexity encoder. Another important characteristic of CM-TSVQ + FEC is its robustness to channel mismatch. CM-TSVQ has a lower rate of decrease in PSNR when the channel cross-over probability is increased than does COVQ — Table 3 presents the values of PSNR for three channels transition probabilities (the quantizers were designed for the $P_e = .001$ channel). As can be seen, the decrease in PSNR is lower than COVQ indicating that the proposed scheme would perform better in a fading, time-varying channel. Our results also show that the LBG + FEC is a wasteful scheme and performs poorly, in general, as compared to both the CM-TSVQ and COVQ.

### 5. CONCLUDING REMARKS

We have shown in this work that the performance of CM-TSVQ can be significantly improved with UEP. Performance better than COVQ was observed. Subjective evaluation of the performance also points the CM-TSVQ as a better scheme.

We are currently investigating the problem of code rate allocation among the layers of the quantizer as well as the use of coding schemes other than linear block codes, e.g., RCPC codes with soft decision.

Other forms of processing are also being investigated.

### 6. REFERENCES


## Table 2. Performance over a BSC at 1.5bpp, $P_e = 10^{-3}$, for a $512 \times 512$ RSS image.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Specifications</th>
<th>PSNR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>COVQ</td>
<td>$3 \times 2$ 9 bits per cell</td>
<td>41.18</td>
</tr>
<tr>
<td>Scheme 1. R: $H^{1\ldots5}$</td>
<td>$CM-TSVQ (P_e = 0.18)$, 13-bit codebook</td>
<td>40.99</td>
</tr>
<tr>
<td>G: $H^{1\ldots2}$</td>
<td>12-bit codebook</td>
<td>40.99</td>
</tr>
<tr>
<td>B: $H^{1\ldots2}$</td>
<td>8-bit codebook</td>
<td>40.99</td>
</tr>
<tr>
<td>Scheme 2. R: $H^{1\ldots4}$</td>
<td>$CM-TSVQ (P_e = 0.18)$, 13-bit codebook</td>
<td>40.91</td>
</tr>
<tr>
<td>G: $H^{1\ldots3}$</td>
<td>12-bit codebook</td>
<td>40.91</td>
</tr>
<tr>
<td>B: $H^{1\ldots2}$</td>
<td>8-bit codebook</td>
<td>40.91</td>
</tr>
<tr>
<td>Scheme 3. R: $H^{1\ldots5}$</td>
<td>$CM-TSVQ (P_e = 0.18)$, 13-bit codebook</td>
<td>40.99</td>
</tr>
<tr>
<td>G: $H^{1\ldots3}$</td>
<td>12-bit codebook</td>
<td>40.99</td>
</tr>
<tr>
<td>B: $H^0$</td>
<td>8-bit codebook</td>
<td>40.99</td>
</tr>
<tr>
<td>Scheme 4. R: $H^{1\ldots3}$</td>
<td>$CM-TSVQ (P_e = 0.18)$, 13-bit codebook</td>
<td>40.93</td>
</tr>
<tr>
<td>G: $H^{1\ldots2}$</td>
<td>12-bit codebook</td>
<td>40.93</td>
</tr>
<tr>
<td>B: $H^0$</td>
<td>8-bit codebook</td>
<td>40.93</td>
</tr>
</tbody>
</table>

Table 3. PSNR (dB) results under channel mismatch

<table>
<thead>
<tr>
<th>Lena</th>
<th>RSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_e$</td>
<td>.001</td>
</tr>
<tr>
<td>COVQ</td>
<td>32.54</td>
</tr>
<tr>
<td>Proposed</td>
<td>33.58</td>
</tr>
</tbody>
</table>


