# Evaluation of some reordering techniques for image VQ index compression

António R. C. Paiva and Armando J. Pinho*

Dept. de Electrónica e Telecomunicações / IEETA
Universidade de Aveiro, 3810–193 Aveiro, Portugal
antrafael@ieeta.pt, ap@det.ua.pt

**Abstract.** Frequently, it is observed that the sequence of indexes generated by a vector quantizer (VQ) contains a high degree of correlation, and, therefore, can be further compressed using lossless data compression techniques. In this paper, we address the problem of codebook reordering regarding the compression of the image of VQ indexes by general purpose lossless image coding methods, such as JPEG-LS or CALIC. We present experimental results showing that techniques available for palette reordering of color-indexed images can also be used successfully for improving the lossless compression of images of VQ indexes.

**Keywords:** Image compression, vector quantization, reordering techniques, lossless image coding

## 1 Introduction

Vector quantization (VQ) aims at reducing the redundancy of an image by encoding a block of pixels (or vector) with an index pointing to a similar block (vector) stored in a codebook. Although this process has some loss of information, the analysis of the pixels as a whole eliminates most of the redundancy while keeping the mutual information which is essential to properly reconstruct the image.

By increasing the block size, the quantization process becomes more efficient. However, for blocks greater than $4 \times 4$ pixels the computational complexity becomes high. Thus, still much redundancy exists between VQ indexes. Considering this problem of memoryless VQ, many methods have been proposed to further reduce the bitrate by suppressing this interblock redundancy. Among them, we find predictive VQ [1], finite-state VQ [2], address VQ [3] and conditional entropy coding of VQ indexes (CECOVI) [4]. These techniques, although more effective than memoryless VQ, are generally also much more complex, and some of them requiring large probability tables both at the encoder and decoder, that have to be obtained by training.

In this paper, we address the problem of codebook reordering regarding the compression of the image of VQ indexes by general purpose lossless image coding methods, such as JPEG-LS [5, 6]or CALIC [7]. Therefore, our objective is to create an image of

* This work was supported in part by the Fundação para a Ciência e a Tecnologia (FCT).

(a) (b) (c) (d)

**Fig. 1.** Images of VQ indexes of Lena ($512 \times 512$, 8 bpp, 29.25 dB) obtained with full-search memoryless VQ, and (a) unsorted codebook, (b) codebook energy reordered, (c) codebook reordered with modified Zeng's and (d) Memon's algorithm.

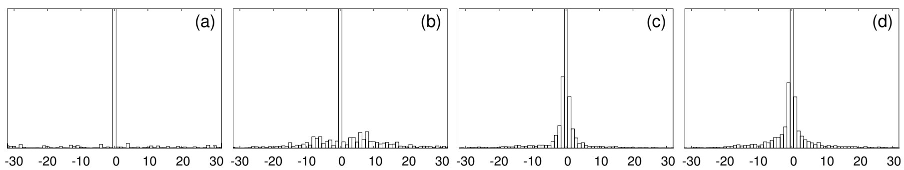


**Fig. 2.** Histograms of the first order differences for the images presented in Fig. 1, respectively. The histograms have been truncated to the $[-32, 32]$ interval to enhance the central region.

VQ indexes that is more appropriate for compression by these techniques, through a suitable reordering of the codebook.

The problem is that, due to the limited block size in VQ still much unnecessary interblock correlation remains to be suppressed. The interblock correlation can be made visible by placing the VQ indexes in an image structure. Looking at the resulting image it is possible to discover some of the original image outline (see Fig. 1(a)). However, the image of VQ indexes usually does not have a Laplacian distribution of the first order prediction residuals as the image coders expect (as can be seen in Fig. 2(a)), which causes the general purpose lossless images coders to perform poorly. Hence, the purpose of image reordering in this context is to reorder the codebook, resulting in an image having a more adequate distribution to lossless compression by the general purpose image coders, therefore, improving the compression results. The effect of reordering can be observed in Figs. 1(b–d) and Figs. 2(b–d).

A similar problem to the one addressed in this paper can be found in coding color-indexed images (a survey can be found in [8], with references to other work on the subject). There, the aim is to reorder the palette of colors such that the reordered image can be better compressed. Given the similarity, we will address some of the methods that have been proposed in that framework, and evaluate how they perform in the VQ context.

The remainder of this paper is organized as follows. In Section 2 we briefly introduce the reordering techniques that we will be evaluating. Section 3 presents the results of simulations on several images. Finally, some conclusions are drawn in Section 4.

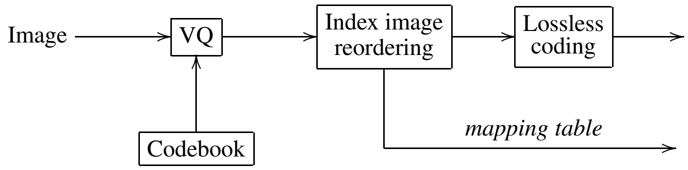


**Fig. 3.** Block diagram of the encoding process

## 2 Reordering techniques

For reordering the images of VQ indexes we will consider one vector-based and two index-based techniques. Vector-based methods try to approximate the distribution to that of the original image which is known to be smooth (for natural images) and with the intended Laplacian distributed differences, and depend only on the codebook that is used. Therefore, needs only to be applied when the codebook is changed. On the other hand, index-based methods analyze the image statistics, and construct a mapping of reduced variance based on these statistics. But, since the index-based techniques greatly depend on the image, a mapping table needs to be transmitted for each image with the information to reverse the reordering.

Figure 3 shows the general case block diagram of the encoding process. In this process the codebook reordering is a preprocessing method aiming to improve the efficiency of the lossless image compression. At the decoder, the process is reversed, by applying the reverse mapping to the image of VQ indexes, before the VQ reconstruction.

### 2.1 Codebook reordering by energy

The concept behind energy reordering is very similar to that of luminance reordering in color-indexed images [9]. The energy of a vector gives us an idea of the vector intensity and, therefore, essentially allows to reconstruct the image contrast structure. Figure 1(b) shows precisely this behaviour, with the image vector quantized with the energy ordered codebook looking very much as a subsampled version of the original image by the block size dimensions (this is, in this case, reduced by a factor of 4 in both directions, since the VQ block size used was $4 \times 4$). With only this very simple step the image histogram of the differences have a different appearance, more similar to a Laplacian distribution, as can be seen in Fig. 2(b). This distribution reflects the new statistics of the image of VQ indexes which translates in a much more efficiently lossless coded image.

The codebook sorting is made by calculating the energy of each vector in the codebook,

$$E = \sum_{i=1}^{M} \sum_{j=1}^{N} v_{ij}^2,$$

where $v_{ij}$ is the pixel at position $(i, j)$ of the vector, considering vectors of $M \times N$ pixels. Then a sorting algorithm is applied to the codebook where the sorting criterion compares the vector energy.

### 2.2 Codebook reordering with the modified Zeng's technique

Zeng *et al.* [10] proposed a re-indexing technique with the the aim of improving the compression of color-indexed images. This technique performs a re-indexing of the indexes independently of their physical meaning, relying only on the statistical information of the image of indexes. Later, a theoretical analysis of Zeng's method by Pinho *et al.* [11] for the case of Laplacian distributed differences conducted to a new set of parameters, different from those originally suggested by Zeng. In the same paper the authors also showed that these proposed parameters provide better results. Since the modified Zeng method depends only on the image statistics, independently of their meaning, it can be used without modification to the reordering of images of VQ indexes. Figure 1(c) shows an image reordered with the modified Zeng technique.

The modified Zeng algorithm performs as follows [11]. The algorithm starts by finding the index that is most frequently found contiguous to other different indexes, and the index most frequently found adjacent to it. These indexes are the initial values of an index list $P_N$, to be constructed iteratively by the algorithm, where $N$ is the number of elements in the list. In each iteration an index is selected from the set of indexes still not in the list and appended to the list either to the right or to the left. We denote by $l_j$ the indexes already in the list and by $u_i$ those still to be assigned. Thus, the initial list is $P_2 = (l_1, l_2)$. In each iteration, the algorithm calculates the index $\sigma$ which satisfies

$$\sigma = \arg\max_{u_i} \sum_{j=1}^{N} C(u_i, l_j),$$

where $N$ is the list size, and $C(i, j)$ is the number of occurrences (on the initial index image) of the index $i$ found adjacent of index $j$. Then, to choose the side of the list where to place the index, we calculate

$$\Delta = \sum_{j=1}^{N} (N + 1 - 2j) C(u_i, l_j).$$

If this expression is positive ($\Delta > 0$) then the index should be placed on the left side of the list, otherwise it should be placed on the right side.

### 2.3 Codebook reordering with Memon's technique

Like the modified Zeng's method described previously, the method by Memon *et al.* [12] is also an index-based reordering algorithm developed in the framework of color-indexed images. Visually (see Fig. 1(d)), the results are similar to modified Zeng, but the image histogram of the differences (Fig. 2(d)) shows some improvements.

As it was referred previously, it is a known fact that the prediction residuals are well modeled by a Laplacian distribution, and that most lossless coders depend on this property to compress the image. Knowing this, the Memon algorithm tries to reorder the indexes so that the prediction residuals are minimized. This corresponds to reducing the zero-order entropy of the residuals. Globally, this is made by the minimization of the sum of the prediction residuals given by

$$\sum_{i=1}^{M}\sum_{j=1}^{M} C(i,j)|i-j|, \tag{1}$$

where $M$ represents the number of indexes. In this case, $C(i,j)$ is the number of times index $i$ is used as the prediction value by a pixel with index $j$. For example, if only first-order causal prediction is used, then $C(i,j)$ is just the number or times index $i$ if found directly after $j$.

The solution to the problem from the perspective presented in Eq. (1) is very difficult, if not impossible. Hence, Memon *et al.* have proposed two heuristics: one based on simulated annealing, and one called "pairwise merging". The "pairwise merging" technique, which we will use, is much faster to compute. Nevertheless, it is computationally more demanding (order $\mathcal{O}(M^4)$) than the modified Zeng's algorithm (order $\mathcal{O}(M^3)$).

In the pairwise merge heuristic, ordered sets of indexes are merged until only one set exists. Through the resulting ordered set one can determine a mapping table to perform the reordering of the image pixels, and therefore, to determine the re-indexed images. Initially, each index is assigned to a different set. Next, the algorithm performs two steps on each iteration. First, the two sets $A$ and $B$ maximizing

$$\sum_{v_i\in A}\sum_{v_j\in B} w(i,j)$$

are selected from all possible pairs of ordered sets, where $w(i,j) = C(i,j)+C(j,i)$. Then, several combinations of the sets $A$ and $B$ are evaluated, and is chosen the one minimizing

$$\sum_{i=1}^{k}\sum_{j=1}^{k} w(u_i,u_j)|i-j|,$$

where $u_1,u_2,\ldots,u_k$ is the ordered set of the vectors from $A$ and $B$ under evaluation. Since the evaluation of all the possible combinations is impractical, Memon *et al.* [12] suggested some combinations that should perform reasonably well in most situations. If $a_1,a_2,\ldots,a_r$ and $b_1,b_2,\ldots,b_s$ are the two ordered sets under evaluation, with $r,s>1$, then the following combinations are considered:

$$\begin{array}{c}
a_1,a_2,\ldots,a_r,\, b_1,b_2,\ldots,b_s\\
a_r,\ldots,a_2,a_1,\, b_1,b_2,\ldots,b_s\\
b_1,b_2,\ldots,b_s,\, a_1,a_2,\ldots,a_r\\
b_1,b_2,\ldots,b_s,\, a_r,\ldots,a_2,a_1
\end{array}$$

Otherwise, if any of the sets has size one, then the following configurations are tested:

$$\begin{array}{c}
a_1,b_1,b_2,\ldots,b_s\\
b_1,a_1,b_2,\ldots,b_s\\
b_1,b_2,a_1,\ldots,b_s\\
\vdots\\
b_1,b_2,\ldots,b_s,a_1
\end{array}$$

## 3 Experimental results

In this Section, we present some compression results of applying general purpose lossless image coders to the VQ index images. The lossless coders JPEG-LS and CALIC were applied over 23 images of VQ indexes, and for each reordering method: energy, modified Zeng (referred in the tables as mZeng) and Memon. For comparison, we have also included the results of the lossless coders applied directly to the images vector quantized before any reordering.

The VQ index images where obtained with a basic full-search memoryless vector quantizer, applied over the 23 images of the kodak images test set[1], with block size $4 \times 4$ (vector dimension 16) and codebook size 256. Each image of the kodak set has dimensions $768 \times 512$, and was not in the codebook training set. To the generation of the codebook we used the generalized Lloyd algorithm [13], also known as LBG algorithm from the article authors, with a training set made up of 13 natural images. The image quality obtained with this VQ method is far from being the best. However, since we are focusing our attention on the increase of efficiency of the lossless compression performed after the reordering, and other VQ methods can be used to enhance the image quality, we believe that this is not really important on the evaluation of the reordering techniques with general purpose lossless coding methods.

First, the whole kodak set was vector quantized using the unsorted codebook that resulted directly from the generalized Lloyd algorithm. Then, the codebook was sorted by energy and a new set of index images was created by vector quantizing again the kodak set. Finally, the two other sets of images were created by applying the modified Zeng and the Memon reordering methods, respectively, to the images of VQ indexes obtained using the unsorted codebook. The implementations of modified Zeng and Memon are the same used in [8] and were provided by the authors of that article. All the other tools where implemented by us.

Table 1 shows the results of image compression, in bits per pixel (bpp), using JPEG-LS[2] and CALIC[3]. The values are for the entire process, i.e., they represent the total number of bits resulting from the lossless compression relative to the number of pixels in the original (before VQ) image. For the index-based methods, which perform the re-indexing for each image, the 256 bytes of mapping table (which depends on the size of the codebook) is also considered. The row labeled "Average" provides overall results for each reordering method. Because of space limitations, we do not present were the tests made with JPEG 2000[4] in lossless mode. The average results for energy, modified Zeng's and Memon's, are 0.346 bpp, 0.299 bpp and 0.294 bpp, respectively.

---

[1] These images are available at `http://www.cipr.rpi.edu/resource/stills/kodak.html` in raster format. The images where then converted to PGM with the Linux utility `rasttopnm`.

[2] This coded can be obtained at `http://www.ece.ubc.ca/spmg/research/jpeg/jpeg_ls/jpegls.html` (version 2.2).

[3] The codec used is available at `http://compression.graphicon.ru/download/i_glless.html`.

[4] This coded is available at `http://www.ece.uvic.ca/~mdadams/jasper/` (version 1.700.5).

**Table 1.** Compression resuls, using a JPEG-LS and CALIC codec, on the kodak image test set. The results for modified Zeng and Memon reordering methods include the mapping table of 256 bytes to be transmitted.

| | JPEG-LS | | | | CALIC | | | |
|---|---|---|---|---|---|---|---|---|
| Image | Unsorted (bpp) | Energy (bpp) | mZeng (bpp) | Memon (bpp) | Unsorted (bpp) | Energy (bpp) | mZeng (bpp) | Memon (bpp) |
| 01 | 0.495 | 0.374 | 0.346 | 0.342 | 0.447 | 0.360 | 0.336 | 0.333 |
| 02 | 0.385 | 0.253 | 0.186 | 0.185 | 0.304 | 0.222 | 0.177 | 0.177 |
| 03 | 0.366 | 0.250 | 0.203 | 0.196 | 0.294 | 0.221 | 0.192 | 0.184 |
| 04 | 0.458 | 0.312 | 0.262 | 0.246 | 0.385 | 0.286 | 0.246 | 0.234 |
| 05 | 0.530 | 0.421 | 0.398 | 0.394 | 0.485 | 0.405 | 0.387 | 0.384 |
| 06 | 0.434 | 0.314 | 0.277 | 0.271 | 0.384 | 0.301 | 0.264 | 0.261 |
| 07 | 0.400 | 0.304 | 0.271 | 0.262 | 0.351 | 0.284 | 0.259 | 0.253 |
| 08 | 0.497 | 0.389 | 0.403 | 0.402 | 0.453 | 0.372 | 0.383 | 0.382 |
| 09 | 0.336 | 0.243 | 0.212 | 0.205 | 0.262 | 0.207 | 0.189 | 0.184 |
| 10 | 0.403 | 0.282 | 0.245 | 0.236 | 0.344 | 0.259 | 0.231 | 0.225 |
| 11 | 0.449 | 0.314 | 0.278 | 0.269 | 0.397 | 0.301 | 0.268 | 0.262 |
| 12 | 0.362 | 0.233 | 0.177 | 0.170 | 0.284 | 0.206 | 0.168 | 0.163 |
| 13 | 0.528 | 0.412 | 0.385 | 0.381 | 0.481 | 0.401 | 0.372 | 0.370 |
| 14 | 0.514 | 0.375 | 0.348 | 0.333 | 0.463 | 0.364 | 0.334 | 0.324 |
| 15 | 0.366 | 0.238 | 0.211 | 0.202 | 0.305 | 0.213 | 0.191 | 0.184 |
| 16 | 0.432 | 0.288 | 0.245 | 0.231 | 0.375 | 0.273 | 0.236 | 0.224 |
| 17 | 0.464 | 0.317 | 0.281 | 0.271 | 0.403 | 0.300 | 0.266 | 0.259 |
| 18 | 0.487 | 0.354 | 0.321 | 0.319 | 0.431 | 0.337 | 0.309 | 0.306 |
| 19 | 0.469 | 0.327 | 0.280 | 0.262 | 0.404 | 0.303 | 0.265 | 0.249 |
| 20 | 0.250 | 0.181 | 0.169 | 0.162 | 0.227 | 0.174 | 0.165 | 0.158 |
| 21 | 0.367 | 0.282 | 0.259 | 0.243 | 0.320 | 0.262 | 0.248 | 0.236 |
| 22 | 0.462 | 0.327 | 0.278 | 0.271 | 0.403 | 0.312 | 0.267 | 0.261 |
| 23 | 0.399 | 0.264 | 0.198 | 0.189 | 0.326 | 0.235 | 0.184 | 0.178 |
| Average | 0.428 | 0.306 | 0.271 | 0.262 | 0.370 | 0.286 | 0.258 | 0.251 |

Not surprisingly, Memon is the reordering method which performs better, for all lossless coders, followed by the modified Zeng. The coding of the index images resulting from a codebook sorted by energy produced the worst results. Nevertheless, the energy reordering allows an average increase of the coding efficiency of 29%, 30% and 23%, comparatively to the unsorted coding, with JPEG-LS, JPEG 2000 and CALIC, respectively. Moreover, all of this is made without any complexity increase at the coding / decoding process since the reordering is made offline. Comparatively to Memon reordering, it has an efficiency decrease of 17%, 18% and 14%, for the same coders. These results of energy codebook reordering are quite interesting since they show how a simple reordering can dramatically improve lossless compression. Comparing modified Zeng with Memon, we see that this method performs only 2–3% worse. Considering these results and the fact that modified Zeng is much faster than Memon, the choice between the two should consider the tradeoff between computational complexity and compression efficiency.

## 4 Conclusions

Reordering techniques provide a simple yet very effective way to improve the compression of images of VQ indexes. In this paper, we evaluated the performance of some reordering techniques when applied to images of VQ indexes that are subsequently losslessly coded, and presented results for three lossless image coders.

Comparing the average results for CALIC and JPEG-LS with Memon reordering with other state-of-the-art VQ coding methods, we see that these are considerably good results. Moreover, this is achieved using only, correctly combined together, *off-the-shelf* algorithms for reordering and lossless image compression.

From the experimental results obtained, we also conclude that Memon's method is the best reordering algorithm for the three image coders. Nevertheless, the modified Zeng's algorithm should also be taken into account since it also performs very good (only a few percent worse than Memon's), but much faster. So, if computational complexity (or time) is a stringent factor, then the modified Zeng's method can provide an efficiency / computation tradeoff very well balanced.

## References


1. Hang, H.M., Woods, J.W.: Predictive vector quantization of images. IEEE Trans. on Communications **33** (1985) 1208–1219
2. Foster, J., Gray, R.M., Dunham, M.O.: Finite-state vector quantization for waveform coding. IEEE Trans. on Information Theory **31** (1985) 348–359
3. Nasrabadi, N.M., Feng, Y.: Image compression using address-vector quantization. IEEE Trans. on Communications **38** (1990) 2166–2173
4. Wu, X., Wen, J., Wong, W.H.: Conditional entropy coding of VQ indexes for image compression. IEEE Trans. on Image Processing **8** (1999) 1005–1013
5. ISO/IEC 14495–1 and ITU Recommendation T.87: Information technology - Lossless and near-lossless compression of continuous-tone still images. (1999)
6. Weinberger, M.J., Seroussi, G., Sapiro, G.: The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS. IEEE Trans. on Image Processing **9** (2000) 1309–1324
7. Wu, X., Memon, N.: Context-based, adaptive, lossless image coding. IEEE Trans. on Communications **45** (1997) 437–444
8. Pinho, A.J., Neves, A.J.R.: A survey on palette reordering methods for improving the compression of color-indexed images. IEEE Trans. on Image Processing (2004 (in press))
9. Zaccarin, A., Liu, B.: A novel approach for coding color quantized images. IEEE Trans. on Image Processing **2** (1993) 442–453
10. Zeng, W., Li, J., Lei, S.: An efficient color re-indexing scheme for palette-based compression. In: Proc. of the 7th IEEE Int. Conf. on Image Processing, ICIP-2000. Volume III., Vancouver, Canada (2000) 476–479
11. Pinho, A.J., Neves, A.J.R.: A note on Zeng's technique for color reindexing of palette-based images. IEEE Signal Processing Letters **11** (2004) 232–234
12. Memon, N.D., Venkateswaran, A.: On ordering color maps for lossless predictive coding. IEEE Trans. on Image Processing **5** (1996) 1522–1527
13. Linde, Y., Buzo, A., Gray, R.M.: An algorithm for vector quantizer design. IEEE Trans. on Communications **28** (1980) 84–95