

# A Codebook Modification Method of Vector Quantization to Enhance Compression Ratio



Dibyendu Barman, Abul Hasnat, and Bandana Barman

**Abstract** The performance of an image compression algorithm is based on the amount of compression ratio achieved keeping the visual quality of the decompress image up to the mark. In an image compression algorithm, two performance measurement parameters, compression ratio, and visual quality of the decompress image are inversely proportional. So, improving the compression ratio of the compression algorithm, keeping visual quality of the decompress image as close to the original is a major challenging task. Vector quantization is one of the widely used lossy image compression techniques found in literature. The compression ratio of this algorithm basically depends on the size of the index matrix and codebook generated during the process. In this present work, a new technique is proposed which represents each and every value of the codebook by 5 bits instead of 8 bits that means it reduces the amount of memory required to store the codebook by 37.50% and which increases the compression ratio of the algorithm significantly. The proposed method is applied on many standard color images found in literature and images from UCIDv.2 database. Experimental results show that the proposed method increases the compression ratio significantly, keeping the visual quality of the decompressed image almost same or slightly lower.

**Keywords** Codebook · Codevector · Compression ratio · Image compression · PSNR · SSIM · Vector quantization

## 1 Introduction

Image compression is a technique that removes the redundancy present in the image that reduces the amount of memory required to store it in a storage medium and also reduces the required bandwidth to transfer the image through a communication

---

D. Barman (✉) · A. Hasnat  
Government College of Engineering and Textile Technology Berhampore WB, Berhampore, India

B. Barman  
Kalyani Government Engineering College, Nadia, West Bengal, India

medium [1–4, 7–9]. The image is basically compressed by taking advantage of three types of redundancies [7, 8]. (i) Coding redundancy: Present in the image when less optimal codeword is used. (ii) Inter-pixel redundancy: Causes due to correlation of the neighboring pixels, (iii) Psycho-visual redundancy-causes due to data ignored by human visual system [8]. Image compression technique broadly classifies into two categories, i. Lossless image compression [8] technique: Visual quality of the decompressed image is the same as original, but the compression ratio achieved is not up to the mark. Normally, this type of compression technique is used where the visual quality of the decompressed image is a major issue rather than compression ratio. Run length encoding (RLE) [7, 8], arithmetic encoding (AE), [7, 8] and Lempel–Ziv–Welch (LZW) [7, 8] are some well-known lossless image compression techniques. ii. Lossy image compression [8]: In the case of lossy image compression, the amount of compression ratio achieved is very high, but at the same time, a huge amount of data is lost due to which the visual quality of the image is degraded. Vector quantization (VQ) [7, 8], color image quantization (CIQ) [7, 8], JPEG [4], and JPEG2000 [4] are some well-known lossy image compression techniques.

The article is organized as follows: Section 2 discusses the literature review. A brief explanation of the proposed method is discussed in Sect. 3. Experimental results are shown in Sects. 4, and 5 concludes the article.

## 2 Literature Review

In literature study, a sufficient amount of code vector matrix or codebook modification process are found. A few of them are discussed below.

In 2019, Abul Hasnat et al. [4] proposed a method where multiple images of same size can be compressed together by combining code vector matrices or codebook of the chrominance channels of all the images in a three-dimensional matrix. This method improves the compression ratio of the algorithm but at the same time degrades the visual quality of the image. In 2018, Rui Li et al. [5] proposed a new general codebook (GCB) design method based on bit rate and distortion which improves the performance of VQ significantly. Pradeep Kumar Shah et al. [6] in 2016 generate codebook by two steps i. The training set is sorted based on the magnitudes of the training vectors. ii. From the sorted list, training vector from every  $n$ th position is selected to form the codevectors. The method reduces the amount of memory required to store the codebook, but at the same time, it degrades the visual quality of the decompressed image.

### 3 Proposed Method

The size of the codebook of vector quantization plays a vital role for measuring the amount of space required to store the compressed image. The aim of this work is to reduce the number of bits required to store the codebook. The proposed method works in two steps (i) Compression: Reduce the size of each value of the codebook to 5 bits from 8 bits. (ii) Decompression: Just reverse process of compression.

#### 3.1 Compression

**Input:** Codebook of vector quantization **Output:** Compressed codebook.

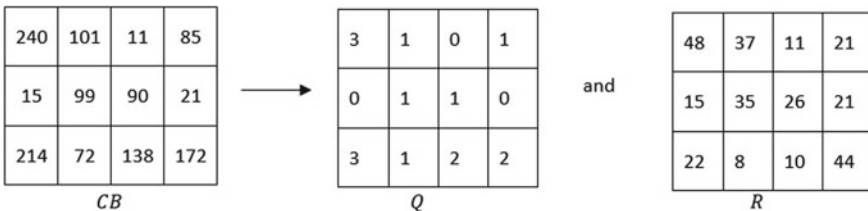
**Step 1:** Let  $CB$  be a codebook of size  $p \times q$ , generated by a normal vector quantization algorithm. The number of elements in the codebook is  $n = p \times q$ .

**Step 2:** Divide each value of  $CB$  by 64, the one fourth value of the maximum value that can be represented by 8 bits, i.e.,  $255/4 = 64$ , and generate the first quotient matrix  $Q$  which contains 0,1, 2, and 3 which can be represented by 2 bits and a remainder matrix  $R$  which contains values between 0 to 63 (Fig. 1).

$$Q_{i,j} = CB_{i,j}/64R_{i,j} = \text{MOD}(CB_{i,j}, 64) \tag{1}$$

**Step 3:** Divide each value of remainder matrix  $R$  by second threshold value 8 and store the nearest integer value in  $Q'$  (Fig. 2).

$$Q'_{i,j} \cong R_{i,j}/8 \tag{2}$$



**Fig. 1** Quotient matrix  $Q$  and remainder matrix  $R$  formation

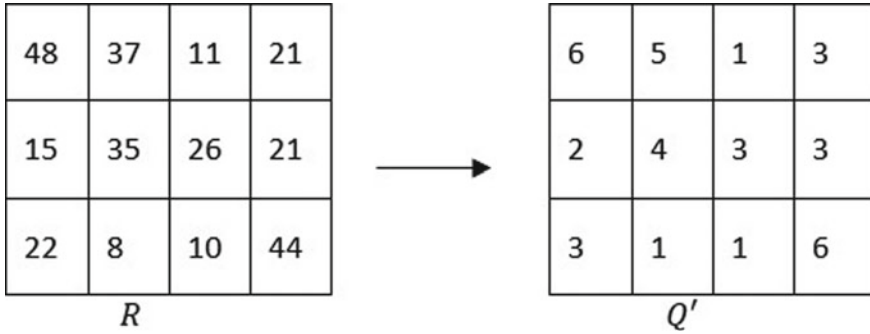


Fig. 2 Second quotient matrix  $Q'$  formation

### 3.2 Decompression

**Input:** Compressed codebook **Output:** Decompressed codebook.

**Step 1:** Multiply each value of  $Q'$  by second threshold value 8 and generate  $Q''$  (Fig. 3).

$$Q''_{i,j} = Q'_{i,j} * 8 \tag{3}$$

**Step 2:** Multiply each value of Quotient matrix  $Q$  of equation no 1 by first threshold value 64 and then add matrix  $Q''$  to generate decompressed codebook  $CB'$  (Fig. 4).

$$CB'_{i,j} = Q_{i,j} * 64 + Q''_{i,j} \tag{4}$$

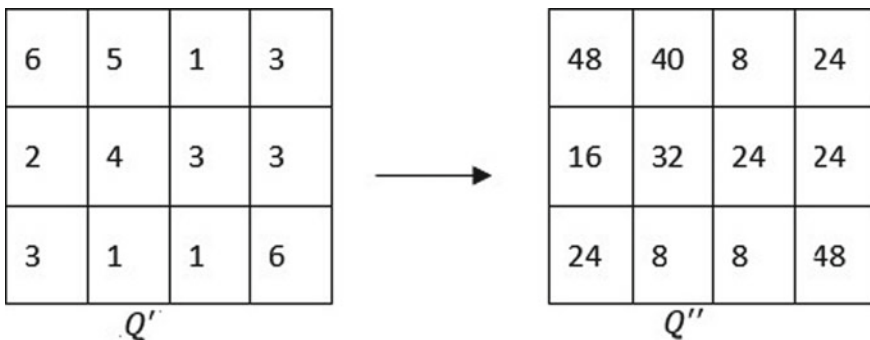


Fig. 3 Creation of  $Q''$  by multiplying  $Q'$  by threshold value 8

240	104	8	88
16	96	88	24
216	72	136	176

 $=$ 

3	1	0	1
0	1	1	0
3	1	2	2

 $*64 +$ 

48	40	8	24
16	32	24	24
24	8	8	48

 $=$ 

240	104	8	88
16	96	88	24
216	72	136	176

$CB'$   $Q*64$   $Q''$

Fig. 4 Decompressed codebook  $CB'$  formation from  $Q$  and  $Q''$

### 3.3 Memory Requirements for Proposed Method

Decompressed codebook  $CB'$  can be recovered by quotient matrices  $Q$  and  $Q'$ . The number of bits required to store the uncompressed codebook  $CB$  is  $12 * 8 = 96$  bits. The maximum value of  $Q$  is 3 and requires 2 bits to represent it, whereas the maximum value of  $Q'$  is 6, i.e., maximum 3 bits are required to store it. So, memory required for  $Q = n * \text{no of bits require to represent MAX element, i.e.} = 12 * 2 = 24$  bits. Similarly, the number of bits required to store  $Q'$  is  $12 * 3 = 36$  bits. Total number of bits require to recover decompressed codebook  $CB' = 24 + 36 = 60$  bits. So, it saves  $(96-60)/96\% = 37.50\%$  memory.

## 4 Experimental Results

The proposed method is implemented using MATLAB2018 and tested on standard color images found in literature and images from UCIDv.2 [4, 7–9] database. The performance of the proposed method is measured by three metrics, compression ratio (CR) [10], peak signal-to-noise ratio (PSNR) [10] and structural similarity index measure (SSIM) [10, 11].

Figure 5a–c shows the original pepper image, decompressed pepper images using vector quantization algorithm and the proposed method, respectively.

From Fig. 5, it can be easily observed that the visual quality of decompressed image using vector quantization and proposed method are almost same. Comparative result of space reduction between vector quantization and proposed method is given in Table 1.

From Table 1, it is observed that space reduction using proposed method lies between 90.81 and 93.81% which is much higher than 88.47–93.23% of VQ.

Table 2 shows computed PSNR [10] between original image and decompressed image using vector quantization and proposed method.

From Table 2, it is observed that visual quality of decompressed image using the proposed method is almost same or slightly lower than VQ in terms of PSNR [10].



**Fig. 5** **a** Original image, **b** decompressed image using vector quantization, **c** decompressed image using proposed method

**Table 1** Space reduction using vector quantization and proposed method

Image name	Space for original image	Vector quantization		Proposed method	
		Total	% Reduction	Total	% Reduction
Girl	196,608	22,659	88.47	18,051	90.81
Jelly beans	196,608	20,767	89.43	16,159	91.78
House	196,608	21,335	89.14	16,727	91.49
Tiffany	786,432	54,097	93.12	49,489	93.70
Baboon	786,432	58,172	92.60	53,564	93.18
Pepper	786,432	53,232	93.23	48,624	93.81
ucid00006	589,824	42,174	92.84	37,566	93.63
ucid00007	589,824	47,456	91.95	42,848	92.73
ucid00008	589,824	41,521	92.96	36,913	93.74

**Table 2** PSNR between original image and decompressed image using vector quantization and proposed method

Image name	Vector quantization			Proposed method		
	Y	Cb	Cr	Y	Cb	Cr
Girl	34.66	38.39	36.87	32.97	37.45	36.57
Jelly beans	37.06	37.98	37.14	34.49	36.72	36.52
House	35.52	37.10	35.48	33.86	36.87	34.53
Tiffany	34.62	34.91	36.80	33.66	34.36	36.45
Baboon	26.04	30.65	31.54	25.16	30.79	31.70
Peppers	33.51	35.17	34.73	32.22	34.95	34.45
ucid00006	25.82	36.11	35.53	24.95	36.49	35.92
ucid00007	25.06	33.76	34.37	23.91	34.52	35.40
ucid00008	28.34	37.65	37.39	27.29	37.45	37.40

**Table 3** SSIM between original image and decompressed image using vector quantization and proposed method

Image name	Vector quantization	Proposed method
Girl	0.8936	0.8569
Jelly beans	0.9793	0.9662
House	0.9553	0.9482
Tiffany	0.9660	0.9591
Baboon	0.8067	0.8020
Peppers	0.9689	0.9657
ucid00006	0.7725	0.7588
ucid00007	0.8374	0.8294
ucid00008	0.8624	0.8381

From Table 3, it can be seen that the quality of the decompressed image using the proposed method slightly lower than vector quantization algorithm in terms of structural similarity index measure (SSIM) [10, 11].

## 5 Conclusion

This study proposes a method which is used to reduce the number of bits require to store the codebook of a vector quantization algorithm. The proposed method represents each value of the codebook by 5 bits instead of 8 bits that mean it reduces the memory required to store the codebook by 37.50% which increases the compression ratio (CR) of the overall algorithm significantly keeping the visual quality of the decompressed image almost same or slightly lower. Future work may be focused on improving the compression ratio more, keeping the visual quality of the image exactly the same as the original.

## References

1. Gonzalez RC, Woods RE, Eddins SL (2011) Digital image processing using MATLAB, Mc-Graw Hill
2. Gan G, Ma C, Wu J (2007) Data clustering theory, algorithms and applications. SIAM (2007)
3. Leitao HAS, Lopes WTA, Madeiro F (2015) PSO algorithm applied to codebook design for channel-optimized vector quantization. IEEE Lat Am Trans 13(4):961–967. <https://doi.org/10.1109/TLA.2015.7106343>
4. Hasnat A, Barman D (2019) A proposed multi-image compression technique. J Intell Fuzzy Syst IOS Press 36(4):3177–3193. <https://doi.org/10.3233/JIFS-18360>
5. Li R, Pan Z, Wang Y (2018) A general codebook design method for vector quantization. Multi Tolls Appl Springer 77(18):23803–23823. <https://doi.org/10.1007/s11042-018-5700-7>
6. Shah PK, Pandey RP, Kumar R (2016) Vector quantization with codebook and index compression In: IEEE International conference system modeling and advancement in research trends, India. <https://doi.org/10.1109/SYSMART.2016.7894488>

7. Hasnat A, Barman D, Halder S, Bhattacharjee D (2017) Modified vector quantization algorithm to overcome the blocking artefact problem of vector quantization algorithm. *J Intell Fuzzy Syst IOS Press* 32(5):3711–3727. <https://doi.org/10.3233/JIFS-169304>
8. Hasnat A, Barman D, Barman B (2021) Luminance approximated vector quantization algorithm to retain better image quality of the decompressed image. *Springer* 80:11985, 12007. <https://doi.org/10.1007/s11042-020-10403-9>
9. Barman D, Hasnat A, Sarkar S, Rahaman MA (2016) Color image quantization using gaussian particle swarm optimization (CIQ-GPSO). In: *IEEE International conference on inventive computation technologies, India*. <https://doi.org/10.1109/INVENTIVE.2016.7823295>
10. Sara U, Akter M, Uddin MS (2019) Image quality assessment through FSIM, SSIM, MSE and PSNR—a comparative study. *J Comput Commun* 7(3):8–18. <https://doi.org/10.4236/jcc.2019.73002>
11. Mandal JK (2020) *Reversible steganography and authentication via transform encoding*. Springer. ISBN: 9789811543975