

# Secure Information Hiding Using Block Match Coding For VQ - Compress Images

Nilam R. Bagul

E&TC Department

Matoshri College of Engineering and Research Centre

Nashik, University of Pune,

India

**Abstract** — Data is very essential in information system, so protecting the data from hackers is very important while transferring the data from one location to another. In this paper, an effective data hiding scheme is proposed which will help in achieving the aim of hiding data into compressed images. These compressed images can be reconstructed without any loss after the secret data is extracted at the decoder. For compressing the images Vector Quantization (VQ) technique is used. In VQ, index modifying and side match VQ (SMVQ) techniques can be used to hide data. Data hiding done by SMVQ technique can provide higher embedding capacity and lower bit rate but it is more time consuming. In contrast, data hiding done by index modifying technique can result into lower embedding capacity and a higher bit rate but it is less time consuming. The proposed scheme will try to merge the advantages of these two techniques while removing the disadvantages of others.

**Keywords** – Vector Quantization; Side Match Vector Quantization; Data hiding; Embedding capacity; Bit rate; Index modifying technique

## I. INTRODUCTION

Representing the data in digital form helps in accessing the data easily and it also increases the efficiency, accuracy and portability of the presented information. The chances of modification of data and violation of copyright increases when some unwanted events occur to access the data. A large amount of data can be transmitted over the internet networks. These network channels are open and very insecure which can lead our secret data to dangerous situations. Today, ensuring that information transmission over the Internet remains safe and secure has become extremely important. To keep the unauthorized user away from the transmission information, a variety of techniques have been proposed; data hiding is one of the protective techniques in data security.

Data hiding can be defined [1] as the process by which a message signal or signature is embedded into a host to get a composite signal. There are three main conflicting requirements of a multimedia data hiding system: perceptual transparency, robustness, and capacity. The composite signal should be perceptually transparent. The data should be recoverable even after the composite multimedia signal has undergone a variety of processing, intentional or unintentional, to remove the embedded data. In other words, the hidden data must be robust against a variety of attacks. It is essential to embed as many bits into the host as possible, or the capacity of the embedding system should be high.

Image compression techniques can be classified into three types [2]: 1) spatial domain methods 2) frequency domain methods 3) Compressed methods. The target of spatial domain compression method is pixels of an image. While, the target of frequency domain compression method is coefficients of a transformed image. Spatial domain compression method is more efficient than frequency domain compression method in terms of computation cost because spatial domain compression method needs not to transform into frequency domain. In other words, spatial domain compression method is more suitable for low power environment (ex. Wireless networks, PDAs). In addition, spatial domain compression methods are usually easily implemented in both software and hardware. The organization of rest of the paper is as follows. Section II explores previous research carried out in data hiding domain. Section III describes the proposed system which includes data embedding and data extraction procedure. Section IV explores the experimentation carried out of the proposed scheme and summarizes the results obtained. Section V finally concludes the paper.

## II. RELATED RESEARCH

The data hiding techniques can be generally classified into two types, reversible data hiding techniques and irreversible data hiding techniques. In irreversible data hiding techniques, only the secret data can be extracted and the cover image can not be recovered at the receiver. While, in reversible data hiding techniques, the secret data can be extracted and the original cover images can also be recovered. In addition, two types of methods, index-modifying and side-match VQ (SMVQ), are normally used to embed secret data into a VQ-compressed image. Many researchers have developed different index-modifying data hiding techniques and SMVQ data hiding techniques. The research carried out by different researchers for data hiding are explained below:

For an SMVQ-based data hiding, Shie *et al.* [8] in year 2006 developed an data hiding technique, based on a Vector quantized compact image, in which blocks are divided into unembeddable and embeddable image blocks based on the differences and side-match distortions (SMDs) [8]. This scheme can achieve a high embedding capacity, but as the quantity of secret data increases the quality of the reconstructed image decreases. Also this scheme is a type of irreversible information hiding.

In year 2007 C. C. Chang, Y. P. Hsieh, and C. Y. Lin, introduced a reversible data hiding method for hiding

confidential information in Vector quantized compact codes based on the declustering approach [11]. The scheme can achieve higher embedding capacity but requires many computations to decluster a codebook into a number of groups.

P. Tsai [12] in the year 2009, proposed the scheme which uses more pairs of peak and zero points in the histogram to achieve the hiding process. Although the scheme is reversible and with formal indices as output, the hiding capacity is rather low, and the distortion of the embedded image increases as the hiding capacity increases.

C. C. Chang, T. D. Kieu, W. C. Wu [13] and J. X. Wang, Z. M. Lu [14] in the year 2009 developed the JNC scheme encodes the encoding index and embeds secret data using the difference between the encoding index and one of its neighboring indices, which can obtain a high embedding capacity and a high BR simultaneously.

K. Hwang and D. Li [18] in the year 2010, developed a character-based trust-organization method enhanced with data coloring (a method of inserting information into covers) and software watermarking, in which data coding and coloring provide solution for maintaining the owner's privacy and data integrity.

In year 2010, C. C. Chen and C. C. Chang [17] proposed a data hiding scheme which used both VQ and SMVQ to hide secret data. In this scheme, when the VQ technique is applied a 1-bit indicator is necessary for every index but only 1-bit confidential information can be implanted in each index. Conversely, when SMVQ technique used more than one bit confidential information can be implanted in each index. The advantage of this technique is that it can offer a high embedding capacity, but it is also an irreversible data-hiding method.

J. M. Guo, J. D. Lee, and Y. H. Chiou [18] in 2011 introduced a data hiding scheme in which SMVQ indices are partitioned into three portions, and indices in portion 1 are used to hide secret data. Although the time-consuming SMVQ is applied in the scheme, it can yield a high embedding capacity and a low bit rate. Moreover, search order coding (SOC)-based and joint neighboring coding (JNC)-based hiding schemes, which use the processed neighboring indices to set up a search path (state codebook) or a reference, were also proposed.

### III. PROPOSED SYSTEM

#### A. DATA EMBEDDING PROCEDURE WITH BMC

During the encoding process, an real image is partitioned into many vectors and by using the table look-up method every vector is encoded by the index of codeword. An index table contains the encoded results. During the decoding procedure, the same generated CB (codebook) is used by the receiver to decode the index back to its respective codeword for reconstructing the original image. An original image is partitioned into 4 blocks of  $2 \times 2$  size. After this each block is then converted to a vector with four dimensions. The block diagram of the proposed system is explained below

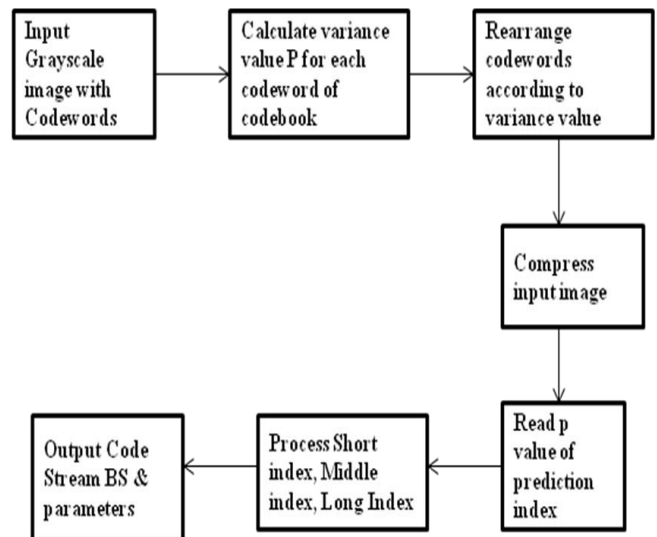


Fig .1. Block Diagram For Data Embedding Procedure

In the proposed scheme, prediction index, is used to predict the encoding index and generate the state codebook. In general, the neighbouring blocks have low correlations with those blocks locating at the extremely non smooth regions, and vice versa. For blocks with high correlations, the prediction index can better predict the encoding index, and vice versa. In the VQ process, when two neighbouring blocks with high correlations are replaced with indices, the correlations may not be existed. To maintain the correlations of the two neighbouring indices, the codewords in the codebook are sorted according to the mean values and variance values of the codewords before the VQ encoding is performed. In the sorting process, only the smooth codewords (with low variances) are sorted using means and variances, while the nonsmooth codewords (with high variances) are not sorted. The value of  $p$  is determined by the variance of a codeword. Given a codeword, if the corresponding variance is greater than a threshold,  $V_{th}$ , the codeword associates to a nonsmooth block and  $p$  is 0. Otherwise, the codeword associates with a smooth block and  $p$  is 1. In the embedding process, the encoding index, namely base index, locating in the first column and first row of a VQ-compressed image remains unchanged. The encoding is done as follows:

1. Divide the image into  $2 \times 2$  non-overlapping blocks of pixels.
2. By making the use of rows of 12 values per pixel window, generate initial cluster of training set.
3. Use any of the algorithm LBG/KPE/KFCG/KMCG for codebook generation on initial cluster to obtain codebook of size 2048 codevectors.
4. Perform dictionary sort on CB.
5. Hide data into sorted CB.
6. Add stego index position column in Stego CB.
7. Sort the stego CB.
8. From sorted CB reconstruct the image to form Stego image.

The flow chart for data embedding procedure is given below :

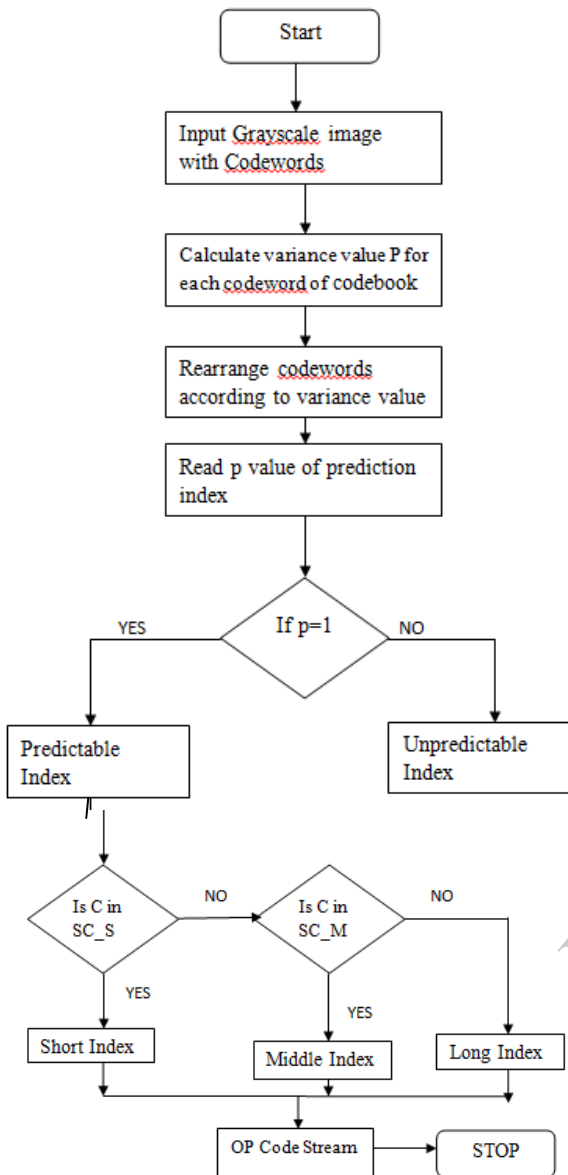


Fig.2. Flow Chart For Data Embedding Procedure

**B. DATA EXTRACTION PROCEDURE**

At the receiver, the decoder can extract the embedded secret data and reconstruct the original VQ indices simultaneously. Initially, the first  $\log_2(cs)$  bits are read from the code stream  $bs$ . The first index that is the base index of the first row of the original VQ-compressed image is restored by the  $\log_2(cs)$  bits. After restoring the base index, each index can be predicted by its corresponding prediction index. The decoding is done as follows:

1. Retrieval of secret message is done using the Stego Image and Stego index position column.
2. The Stego image is partitioned into blocks which generates a set of training vectors. Group of these exclusive training vector generate a codebook.
3. The entries of codebook are arranged using Stego index position column.

4. The secret data is extracted to get back the secret message.

The block diagram for data extraction is explained below

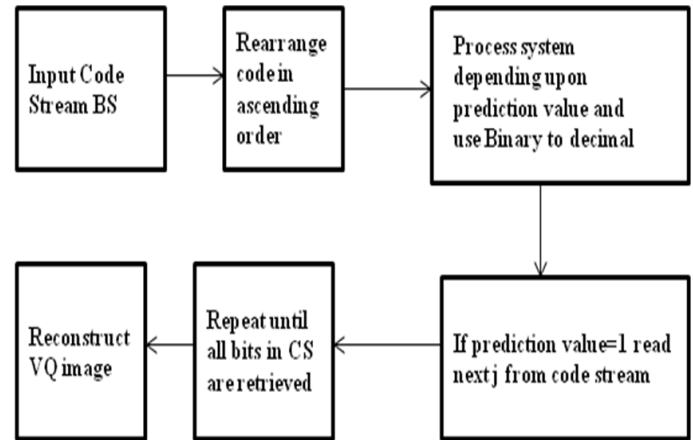


Fig. 3. Block Diagram For Data Extraction Procedure

Let  $rx$  and  $rv$  represent the restored index and the value retrieved from the code stream with length  $\log_2(cs)$  bits. For an unpredictable index, the restored index  $rx$  can be obtained by [14]

$$rx = rv \tag{7}$$

The encoded index can be a predictable index or unpredictable index according to its corresponding prediction value. An unpredictable index can be easily restored. For a predictable index,  $J$  bits converted to a decimal value  $Jv$  are read from the code stream  $bs$ , and then considered as an indicator. Let  $rx$ ,  $rsd$ ,  $rsdv$ ,  $Jv$ ,  $Iv$ , and  $CSv$  represent the restored index, the restored secret data, the values of the retrieved  $s$ ,  $J$ ,  $I$ , and  $\log_2(cs)$  bits from the code stream, respectively. For a predictable encoded index, the restored index  $rx$  and the restored secret data  $rsd$  can be obtained by

$$rx = Jv + Ks \text{ if the encoded index is a short index} \tag{8}$$

$$rsd = rsdv \text{ if the encoded index is a short index} \tag{8.1}$$

$$rx = Iv + Km \text{ if the encoded index is a middle index} \tag{9}$$

$$rx = CSv \text{ if the encoded index is a long index} \tag{10}$$

The flow chart for data extraction procedure is explained below :

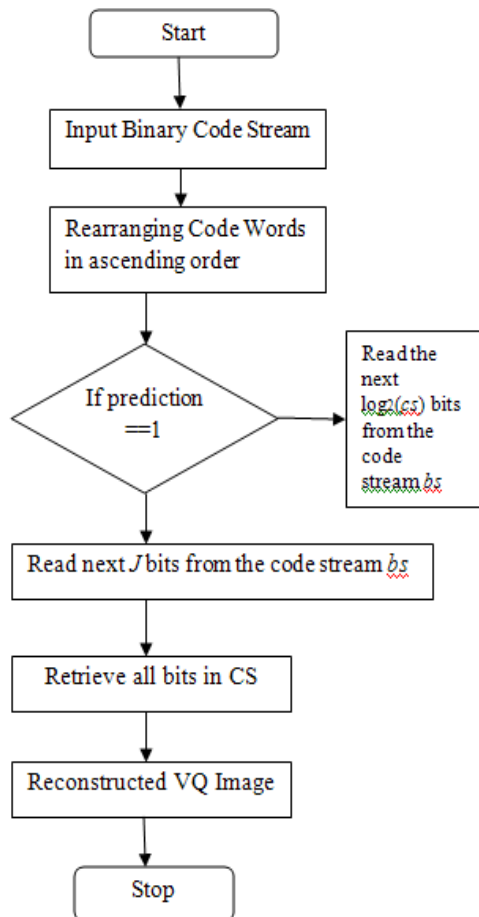
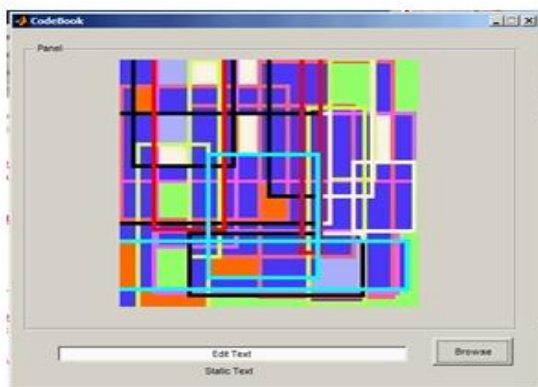


Fig. 4. Flow chart for data extraction procedure

#### IV. EXPERIMENTATION & RESULTS

##### A. Training Vector Generation

Training vectors are required to generate a codebook which will act as a set of training vectors. These training vectors can be used for finding euclidean distance with the codebook generated. The Euclidean distance is required for obtaining the nearest codeword which is helpful in finding the index of respective vector (block). These indexes are necessary for compressing the image. This compressed image is then taken as a input for codebook generation. In our proposed system the test image used for training vector generation is



The training vector generated are :

TABLE 1. GENERATED CODEBOOK

INDEX	CODEVECTORS			
1	84	84	84	84
2	84	84	84	84
3	84	10	84	10
4	10	84	10	84
5	84	84	84	84
6	84	84	84	84
7	84	84	84	84
8	76	76	76	76

After this vector quantization is used for codebook generation and compressing the image.

##### B. Codebook Generation

Our proposed system uses LBG algorithm for VQ codebook generation. For generation of codebook the iteration of LBG is given as follows:

Step 1: Consider an initial codebook  $CB_0$  and randomly generate it.

Step 2: Initially consider  $i = 0$ .

Step 3: The following process is to be carried out for each training vector .

➤ Evaluate the Euclidean distances between the codewords in  $CB_i$  and the training vector. For calculating the Euclidean distance use the below formula

$$d(X, C) = \sqrt{\sum_{t=1}^k (x_t - c_t)^2}$$

➤ From  $CB_i$  find out the nearest codeword.

Step 4: Divide the codebook into  $N$  cells.

Step 5: Generate the new codebook  $CB_{i+1}$  by calculating centroid of every block.

Step 6: Calculate average distortion for  $CB_{i+1}$ . The codebook

converges , if the average distortion is changed by a small amount by the last iteration and after this the process stops.

else,  $i = i + 1$  and go to Step 3.

For an example consider five training vectors to show how to train a codebook. Table 2 shows a set of training vectors. At the initial stage, let the total number of the codewords in a codebook is three, so  $N = 3$ . First, generate an initial codebook  $CB_0$  randomly. This initial codebook is shown in table 2. Secondly calculate the Euclidean distances between the codewords of  $CB_0$  and the set of training vectors. Table 3 concludes that for  $i = 0$  the nearest code of  $X_1$  is  $C_3$ . Group the training vectors into the same cell, which have the similar nearest codeword. This method helps in finding the centroid of every cell to acquire a new codebook. In the specified example,  $X_1$  and  $X_4$  that have the similar nearest codeword  $C_3$  are grouped together into the same cell. The centroid of the two vectors is the third codeword of the

new generated codebook *CB1*. This process is repeated until the codebook is obtained. The final codebook generated is *CB3*.

TABLE 2. SET OF TRAINING VECTORS

	$X_1$	$X_2$	$X_3$	$X_4$
$X_1$	241	192	21	156
$X_2$	212	76	123	36
$X_3$	10	220	108	233
$X_4$	165	108	155	41
$X_5$	109	52	19	247

TABLE 3. CODEBOOK GENERATION USING LBG ALGORITHM

X	Nearest Codeword	Euclidean Distance			Codebook				
		$C_1$	$C_2$	$C_3$	$C_1$	32	177	143	210
$X_1$	$C_3$	248.41	225.942	216.959	$C_2$	196	16	46	24
$X_2$	$C_2$	270.7	99.6444	126.831	$C_3$	180	130	212	101
$X_3$	$C_1$	63.93	351.763	255.421					
$X_4$	$C_3$	226.17	146.952	86.9368					
$X_5$	$C_1$	195.7	243.563	263.989					

In VQ, there are three main steps specifically, codebook generation, encoding process and decoding process. In the codebook generation process, different images are partitioned into many *k*-dimension training vectors. By using the clustering techniques the codebook is generated from the training vectors. In the encoding procedure, by using a table look up method an original image is divided into many vectors which are *k* dimensional and the index of every codeword is used to encode every vector. The encoded results form an index table. During the decoding process, the same codebook is used by the receiver to convert the index back to its corresponding codeword for reconstructing the image.

C. Comparison Of Proposed System With Different Techniques

To present the improvement of our proposed system, the proposed method is compared with the VQ-based scheme [10], the JNC-based schemes [13] and the state-codebook based (SOC and SMVQ) schemes [15], [4] Table 4 shows the performance comparisons of Yang and Lin's [15] scheme, Chang *et al.*'s [10] scheme, and the proposed scheme. Compared to Chang *et al.*'s [10] scheme, the proposed scheme can achieve a greater ER and a smaller BR for each image. the scheme proposed by Chang *et al.* [10] was extended by Yang and Lin [15] and achieved a better performance in terms of EC and ER by incorporating both of the SMVQ and VQ techniques. The execution time of the embedding and the extraction procedures for the proposed scheme and the former schemes are organized in Table 4.

TABLE 4. COMPARISON WITH DIFFERENT TECHNIQUES

Type of image	Average Execution Time (ms)							
	Chang et al's scheme		Chang et al's scheme		Wang & Lu's scheme		Proposed scheme	
	256	512	256	512	256	512	256	512
Color Image	1.06	1.16	8.97	5.97	1.20	1.05	1.01	0.89
Gray image	1.14	1.24	8.84	5.78	1.15	0.96	0.89	0.85

The table with codebook of size 256 shows that the proposed scheme can achieve the fastest average execution time, followed by Yang and Lin's [16] scheme (VQ), and then Chang *et al.*'s [12] scheme. Similarly, when codebook is of size 512, the proposed scheme has the fastest processing efficiency, followed by Wang and Lu's [13] path 1 scheme, and then Chang *et al.*'s [10] scheme. As it can be seen, the execution speed of Chang *et al.*'s [19] scheme and Yang and Lin's [16] scheme (VQ+SMVQ) are rather slow. In Chang *et al.*'s [4] case, it is caused by a required search path for each encoding index, leading to an additional time cost. In Yang and Lin's [16] scheme (VQ+SMVQ), it spends more time on computing the SMD values to setup the state codebook for the encoding index. The comparison table concludes that the proposed system have less execution time than the previous proposed methods. The graphical representation of the results is plotted below

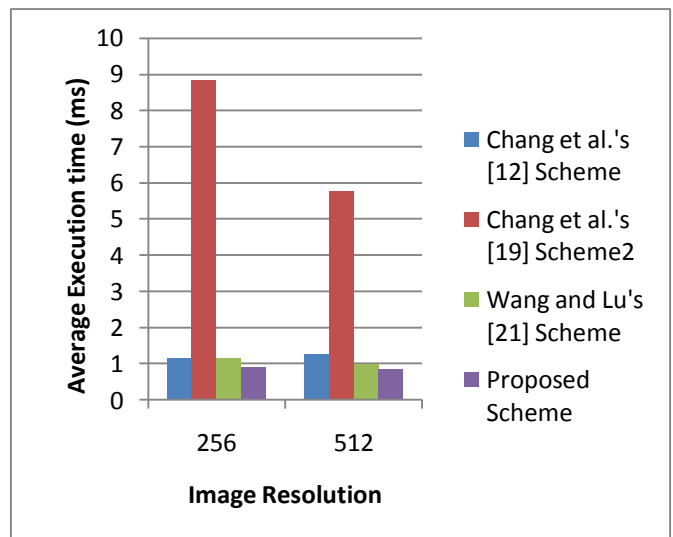


Fig. 4. Execution Time for Color Images

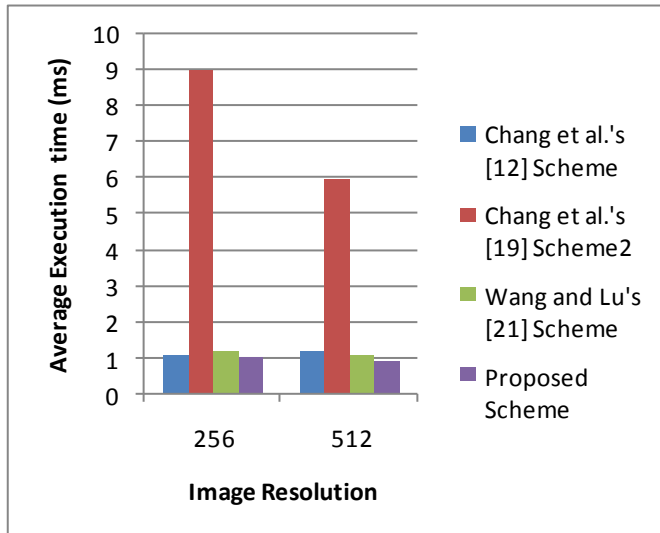


Fig. 5. Execution Time for Grayscale Images.

From these graphs it is clear that the execution time of our proposed system for colour and gray scale images is minimum that the already available systems.

## V. CONCLUSION

The proposed method will combine the advantages of both index modification vector quantization (lower embedding capacity, higher bit rate, and less time consuming) and side match vector quantization (higher embedding capacity, lower bit rate, and more time consuming). Compared with the former schemes, the proposed scheme has the following characteristics:

- The original cover image can be recovered completely after the secret data extraction.
- This approach can provide a very high embedding capacity, very high embedding rate, and limited time consuming compared to the SMVQ.
- A state codebook is efficiently generated by the prediction index, and thus the encoding/hiding can be easily accomplished using a simple calculation rather than a table lookup.

## REFERENCES

- [1] M. Jo and H. D. Kim, "A digital image Water marking scheme based on vector quantization," *IEICE Trans. Informat. Syst.*, vol. E85-D, no. 6, pp. 1054–1056, 2002.
- [2] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 8, pp. 890–896, Aug. 2003.
- [3] L. M. Cheng, and C. K. Chan "Hiding data in images by simple LSB substitution," *Pattern Recogn.*, vol. 37, no. 3, pp. 469–474, 2004.
- [4] G. M. Chen, C. C. Chang, and M. H. Lin, "Information hiding based on search-order coding for VQ indices," *Pattern Recognit. Lett.*, vol. 25, no. 11, pp. 1253–1261, 2004.
- [5] K. C. Fan, C. L. Tsai, H. F. Chiang, and C. D. Chung, "Reversible data hiding and lossless reconstruction of binary images using pair-wise logical computation mechanism," *Pattern Recognit.*, vol. 38, no. 11, pp. 1993–2006, 2005.
- [6] N. Ansari, Z. Ni, Y.-Q. Shi, and W. Su, "Reversible data hiding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 3, pp. 354–362, Mar. 2006.
- [7] S. D. Lin, S. C. Shie, and C. M. Fang, "Adaptive data hiding based on SMVQ prediction," *IEICE Trans. Informat. Syst.*, vol. E89-D, no. 1, pp. 358–362, 2006.
- [8] C.Y.Lin and C.C.Chang, "Reversible steganography for VQ compressed images using side matching and relocation," *IEEE Trans. Inform. Forensics Security*, vol. 1, no. 4, pp. 493–501, Dec. 2006.
- [9] W. C. Wu, C. C. Chang, and Y. C. Hu, "Lossless recovery of a VQ index table with embedded secret data," *J. Visual Commun. Image Representation*, vol. 18, no. 3, pp. 207–216, 2007.
- [10] C. Y. Lin, C. C. Chang, and Y. P. Hsieh, "Lossless data embedding with high embedding capacity based on declustering for VQ-compressed codes," *IEEE Trans. Informat. Forensics Security*, vol. 2, no. 3, pp. 341–349, Sep. 2007.
- [11] P. Tsai, "Histogram-based reversible data hiding for vector quantization compressed images," *IET Image Process.*, vol. 3, no. 2, pp. 100–2114, 2009.
- [12] C. C. Chang, T. D. Kieu, and W. C. Wu, "A lossless data embedding technique by joint neighboring coding," *Pattern Recognit.*, vol. 42, no. 7, pp. 1597–1603, 2009.
- [13] J. X. Wang and Z. M. Lu, "A path optional lossless data hiding scheme based on VQ joint neighboring coding," *Informat. Sci.*, vol. 179, no. 19, pp. 3332–3348, 2009.
- [14] Y. Linde, A. Buzo and R. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. 28, no. 1, pp. 84–95, Jan. 2010.
- [15] T. Kim, "Side match and overlap match vector quantizers for images," *IEEE Trans. Image Process.*, vol. 1, no. 2, pp. 170–185, Apr. 2010.
- [16] C. C. Chang and C. C. Chen, "High capacity SMVQ-based hiding scheme using adaptive index," *Signal Process.*, vol. 90, no. 7, pp. 2141–2149, 2010.
- [17] K. Hwang and D. Li, "Trusted cloud computing with secure resources and data coloring," *IEEE Internet Computer.*, vol. 14, no. 5, pp. 14–22, Sep./Oct. 2010.
- [18] Y. H. Chiou, D. Lee, and J. M. Guo, "Reversible data hiding based on histogram modification of SMVQ indices," *IEEE Trans. Informat. Forensics Security*, vol. 5, no. 4, pp. 638–648, Dec. 2011.