A VQ-DCT Scheme for Grayscale Image Watermarking

A. K. Pal, S. Das, G.P. Biswas and S. Mukhopadhyay

Department of Computer Science and Engineering,
Indian School of Mines, Dhanbad-826004, India

ABSTRACT
In this paper, a hybrid watermarking scheme, that employs both DCT and VQ, has been presented. The authors have proposed a modification of the conventional LBG algorithm in this paper for designing the codebook to be used subsequently for compressing the image employing VQ. The watermark/logo is embedded in the constructed codebook after suitable DCT operations. The compression and reconstruction of the cover image employs the modified embedded codebook. The indirect mechanism of embedding the watermark and the hybrid combination of VQ and DCT make the entire watermarking scheme reasonably robust with respect to various types of common attacks which include intensity modification in real domain, modification in transformed domain namely JPEG compression etc. The method has been implemented and tested on a number of example images. We have also tested the robustness of authenticating capability of the scheme both qualitatively and quantitatively for various types of attacks.

General Terms
Attacks, Image Compression, Watermarking.

Keywords
Digital Watermarking, DCT, LBG, Vector Quantization.

1. INTRODUCTION
Usage of digital image watermarking technique [1] has grown significantly to protect the copyright ownership of multimedia data. Since Digital multimedia data is very much prone to unlawful and unauthorized replication, reproduction and manipulation. Digital watermarking technique is one of the popular and widely used copyright authentication techniques for multimedia data. In digital image watermarking scheme, copyright protection information which is commonly termed as watermark is embedded in the protected image. The watermark may be a logo, label or a random sequence. A typical good watermarking scheme should aim at keeping the embedded watermark very robust under malicious attack in real and spectral domain.

Incorporation of the watermark in the image could be performed in various ways [2-4]. In addition to this incorporation could also be realized in transformed version of the protected image [5-9]. For example, M. Barni *et.al* [5] was presented a digital watermark scheme based on DCT transformed image where they embedded a random sequence into the selective coefficients of DCT transformed image. Another DCT based watermark scheme was proposed by Wai C. Chu [6]. In their scheme, before embedding the watermark, DCT applied on sub images, which were obtained by sub sampling the image. Jiang-Lung Liu *et.al* [7] proposed a robust watermark scheme by embedding a watermark into the DWT domain. Some combinational approach [8-10] of two transforms namely DCT and DWT were used to design robust watermarking scheme.

Basically, in all scheme, the incorporation of the watermark is performed directly either in image or transformed image. Besides some indirect incorporation methods are available where embedding is done during encoding process. For example, in VQ based watermark scheme [12-15], instead of inserting the logo directly into the protected image, the embedding is done during VQ encoding process. The main advantage of their algorithms is that in general these are more robust compare to the direct method. Since when attacks are employed directly on the watermark image, it does not damage the watermark directly.

In this work, we try to make this scheme more robust which is hybrid in nature and employ both image transform namely VQ and DCT. Further in our scheme, the watermark exists in both the VQ compressed image as well as in the VQ decoded image and the watermark extraction can be performed without using the original image.

The paper is organized as follows. After this introductory section, we present a brief overview of the VQ and the 2D-DCT in section 2. The proposed digital watermarking scheme is elaborated in section 3. Some experimental results are presented in section 4 to discuss the robustness of the proposed scheme with respect to several image processing attacks. Finally, conclusions are given in section 5.

2. BRIEF CONCEPT OF VQ AND DCT
In this section, we will briefly review the VQ and the DCT in the context of digital watermarking scheme. VQ extensively use for compressing the image with high compression ratio while DCT is widely used to implement robust digital watermarking algorithm.

2.1 VQ for Image Compression
In conventional Vector quantization (VQ) scheme [16-17], gray level image is decomposed into blocks of size $p \times q$. Then each block is lexicographically converted into a vector of dimension, $k (=pq)$. In VQ process, a group of similar vectors are represented by a representative vector. This representative vector is called codeword and the set of all codewords is called codebook, which is basically a look up table which contains codewords along with their indices i.e. location in the lookup table/codebook arranged in some prescribed sequence. Basically, there are three major steps in the VQ process namely (i) codebook design, (ii)

encoding process and (iii) decoding process. Among several algorithms for codebook design, one of the widely use technique is *LBG* algorithm [18]. The *LBG* algorithm begins with choosing some training vectors at random as an initial codebook. This initial codebook is evolved into an improved one through iterative clustering of the training vectors. The algorithmic steps of *LBG* are as given below.

***Algorithm:* LBG_Codebook_Design**

Input: A training set, $X = \left\{ x_i \middle| x_i \in R^k, \forall i = 1,2,...,N \right\}$

Output: A codebook, $Y = \left\{ y_j \middle| y_j \in R^k, \forall j = 1,2,...,M \right\}$

Begin

Step 1: Initialize the codevectors of Y by selecting the training vectors at random from the training set X.

Step 2: Add each vector, from the training set X, towards a cluster, say C_j in such a way that $j = Min\{ d(x_i,y_j)\}$ where $j=1, 2, 3, ..., M$. $x_i \in X$, $y_j \in Y$ and $d(x_i, y_j)$ is denoted as the squared Euclidean distance.

Step 3: Calculate the average distortion for each cluster by the following equation, $D_j = \frac{1}{m}\sum_{i=1}^{m} d\left(x_i, y_j\right)$ where C_j is the j-th cluster having m number of training vectors and $x_i \in C_j$. Here y_j denotes the codeword of cluster C_j.

Step 4: Evaluate the overall distortion for M codewords by the following equation $D_{ov} = \frac{1}{M}\sum_{i=1}^{M} D_i$ where D_i denotes the average distortion of i-th cluster.

Step 5: If $\left|\frac{D_{ov} - D_{ov-1}}{D_{ov}}\right| \le \varepsilon$ stop the iteration, where ε is a threshold given in advance.

Else update the codeword from the centroid of each cluster, C_j and go to *Step 2* for next iteration.

End

After designing the codebook, the VQ encoding process is performed to compress the image. During the VQ encoding process each image vector is replaced by the index of the most appropriate representative vector from the codebook. The matching is obtained based on the smallest squared Euclidean distances between the image vector and the codeword of the codebook.

In case of decoding, each index is used to search the same codebook and the corresponding codeword is placed in the position indicated by the index to get the decompressed image. The schematic diagrams of VQ encoding and decoding process are shown in *Figure 1-2*.

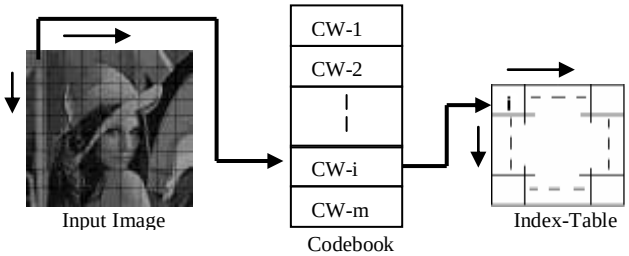


Figure 1: VQ Encoding Process

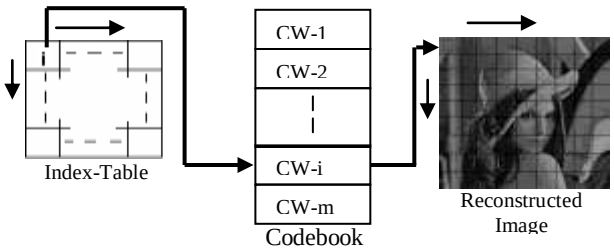


Figure 2: VQ Decoding Process

Where CW⇨ Codeword

2.2 DCT Transformation

The DCT [11] transforms a signal from a spatial representation into a frequency representation. It represents an image as a sum of sinusoids of varying magnitudes and frequencies. An input image, $f(x,y)$ of size $N \times N$ pixels is divided into non-overlapping blocks of size 8×8 and then each block is transformed into DCT coefficients. Two dimensional discrete cosine transform (2D-DCT) for image $f(x,y)$ of size $N \times N$ pixels, is defined as follow.

$$F(u,v) = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\alpha(u)\alpha(v)\cos\left[\frac{(2x+1)u\pi}{2N}\right]\cos\left[\frac{(2y+1)v\pi}{2N}\right] \quad (1)$$

Where $\alpha(u) = \begin{cases} \sqrt{1/N} & for\ u = 0 \\ \sqrt{2/N} & for\ u = 1,2,\ldots,N-1 \end{cases}$

The image is reconstructed from the transformed image $F(u,v)$ by applying inverse 2D-DCT according to the following equation.

$$f(x,y) = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1} F(u,v)\alpha(u)\alpha(v)\cos\left[\frac{(2x+1)u\pi}{2N}\right]\cos\left[\frac{(2y+1)v\pi}{2N}\right] \quad (2)$$

After applying 2D-DCT on digital image, the transform gives three frequency sub-bands: low frequency sub-band, mid-frequency sub-band and high frequency sub-band. Low frequency sub-band contains most of the important visual parts of the image and in general the high frequency sub-band is usually removed by compression. In this paper we have chosen the mid-frequency sub-band and the watermark is embedded by modifying the coefficients of this frequency sub-band so that the visual quality of the image does not be degraded and the watermark will not be removed by compression technique [5].

3. THE PROPOSED TECHNIQUE

In this section, we describe the proposed watermarking scheme which is employed on VQ compressed image. In our scheme, initially VQ technique is applied on the original image and after VQ process it produces a codebook and an index-table. So after VQ process, the codebook is considered as a cover medium for embedding the watermark (Binary image). After embedding the watermark in the codebook, the image can be stored/transmitted either in compressed form (i.e. the watermark embedded codebook and the index table) or in VQ reconstructed watermark image (i.e. reconstructed image after VQ decoding process using the watermark embedded codebook and the index table).

When the image is stored/transmitted as compressed form, then the watermark extraction can be performed directly from the watermark embedded codebook. In the second case, when the watermark image is in from of VQ reconstructed image, then one issue arises during watermark extraction process. The issue is how to extract the watermark embedded codebook from the VQ reconstructed image? So we have proposed a method to extract the codebook from the VQ reconstructed image. First of all we present the codebook extraction method and then we describe the watermark embedding as well as the watermark extraction processes in details in the following subsections.

3.1 Codebook Extraction:

In our scheme, the proposed modified LBG algorithm is used to design the codebook. The reason behind choosing the modified LBG algorithm will be elaborated in details after presenting the algorithmic steps of the modified LBG. In the conventional LBG algorithm the codebook, *Y* is initialized by the randomly selected some training vectors from the training set *X*. In the proposed scheme, instead of initializing the codebook randomly, we have initialized the codebook from the training set *X* based on some pseudo random sequence. The pseudo random sequence will be generated by using a PRNG and a seed. The algorithmic steps of the modified LBG are as given below.

Algorithm: **Modified_LBG_Codebook_Design**

Input: A training set, $X = \left\{ x_i \middle| x_i \in R^k, \forall i = 1,2,...,N \right\}$

Output: A codebook, $Y = \left\{ y_j \middle| y_j \in R^k, \forall j = 1,2,...,M \right\}$

Begin

Step 1: Select a PRNG and a seed. Using the PRNG and the seed generate a pn-sequence, $PN = \{b_1, b_2, \ldots, b_M\}$

Step 2: Initialize the codebook, $Y = \left\{ y_i \middle| y_i \in x_{b_i}, \forall i = 1,2,\ldots,M \right\}$

Step 3: Add each vector, from the training set *X*, towards a cluster, say C_j in such a way that $j = Min\{ d(x_i, y_j) \}$ where $j = 1, 2, 3, \ldots, M$. $x_i \in X$, $y_j \in Y$ and $d(x_i, y_j)$ is denoted as the squared Euclidean distance.

Step 4: Calculate the average distortion for each cluster by the following equation, $D_j = \frac{1}{m} \sum_{i=1}^{m} d\left(x_i, y_j\right)$ where C_j is the *j*-th cluster having *m* number of training vectors and $x_i \in C_j$. Here y_j denotes the codeword of cluster C_j.

Step5: Evaluate the overall distortion for M codewords by the following equation $D_{ov} = \frac{1}{M} \sum_{i=1}^{M} D_i$ where D_i denotes the average distortion of i-th cluster.

Step 6: If $\left| \frac{D_{ov} - D_{ov-1}}{D_{ov}} \right| \leq \varepsilon$ stop the iteration, where ε is a threshold given in advance.
Else update the codeword from the centroid of each cluster, C_j.

Step 7: Replace $x_{b_i} = y_i$ where y_i is the updated codeword obtained from Step 6. Go to Step 2 for the next iteration.

End

In the modified LBG algorithm, during the first iteration at step 3, the image block at b_i-th location gives minimum Euclidean distance with i-th codeword as they are same image block. Through the second iteration the image block at b_i-th location will also match with i-thcodeword since in the first iteration at step 7 the b_i-th location image block was replaced with the i-thcodeword. The above processes imply that the image block at b_i-th location always matched with the i-th codeword during each iteration. Therefore once the codebook is prepared to use, we can compress the secret images by the VQ scheme to generate the index table. In the index-table, we always find the index-value at b_i-th location is *i*. The advantage of using this index-table is that from VQ reconstructed image if we decompose the VQ reconstructed image into non-overlapping blocks then the image block at b_i-th location represents the i-thcodeword and so on. So once we generate the pseudo random sequence, we can easily extract the codewords from the VQ reconstructed image.

3.2 Watermark Embedding Process

When the image compression has been completed by the VQ scheme then we can start embedding the watermark. The watermark embedding procedure is shown in *Figure. 3,* and described in details in the following steps.

Begin

Step 1: Decompose the grayscale image into non-overlapping blocks with size of *8×8.*

Step 2: Use the modified *LBG* algorithm to construct the codebook. The codeword size must be *8×8.* Compress the image using the codebook and store the index-table.

Step 3: Shuffle the position of the codewords within the codebook according to the pseudo random sequence, *pn1* which is generated using the secret seed. This step strengthened the confidentiality of the watermark.

Step 4: Compute block based DCT of each codeword and select *n* coefficients from the mid-frequency sub-band of each DCT transformed codeword.

Step 5: Embed each bit of watermark, as pseudo random noise sequence of length *n* into each DCT transformed codeword as follow.

Step 5.1: Generate a pseudorandom sequence, *pn2*. Here the sequence is in normal distribution with zero mean and unity variance.

Step 5.2: Select the value of α, where α controls the embedding strength. Now if t_i represents the vector of selected mid-band coefficients of i-th codeword, then modify the mid-band coefficients by the following equation to embed the watermark bit as follows:

If the watermark bit is 1 then

$$u_i = t_i + \alpha \left| t_i \right| pn2$$

Else $u_i = t_i$

End If

Step 6: Replace the mid-band coefficients of the DCT transformed codeword by the modified mid-band coefficients from the previous step.

Step 7: Compute inverse DCT to each codeword to produce the watermark codebook.

Step 8: Perform inverse shuffling of the codebook using the pn-sequence from Step 3.

Step 9: Store/transmit the watermark image either in compressed form (i.e the watermarked codebook and the index-table) or in VQ reconstructed form (i.e. reconstructed image after performing VQ decoding using the watermarked codebook and the index-table).

End

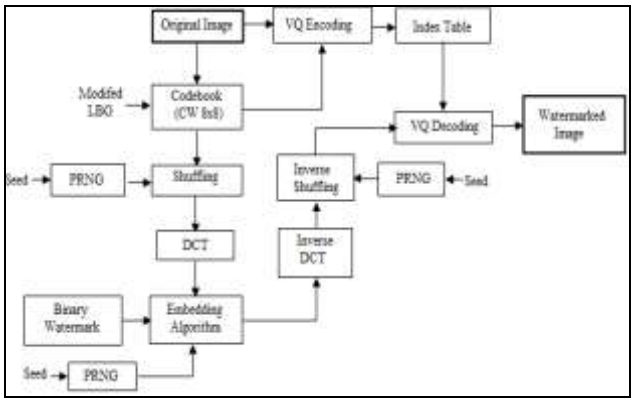


Figure 3: Block diagram of the proposed watermark embedding process

3.3 Watermark Extraction Process

In this subsection, the watermark extraction process is described in details. *Figure 4* shows the block diagram of the watermark extraction process. The algorithmic steps of the extraction process as follow.

Begin

Step 1: If the watermark is extracted from the VQ reconstructed image go to Step 2, otherwise go to Step 3.

Step 2: Decompose the watermark image into non-overlapping blocks of size *8×8* and select the image blocks according to the pn-sequence, which was used during codebook training process. The selected image blocks are store in prescribed order to produce the codebook.

Step 3: Generate the pn-sequence, *pn1* to shuffle the codebook.

Step 4: Compute block based DCT of each codeword and choose *n* coefficients from the mid-frequency sub-band of the each DCT transformed codeword where t_i represents the vector of selected mid-band coefficients of i-th codeword.

Step 5: Find the presence of noise sequence, *pn2* in each codeword to extract the watermark bit as follows.

Step 5.1: Compute the correlation, Z between the t_i and pn2 is defined as:

$$Z = \frac{t_i . pn2}{n}$$

Step 5.2: Estimate the threshold value, $S_z = \frac{\alpha}{3n}\sum_{i=1}^{n}|t_i|$

Step 5.3: If ($Z \geq S_z$) then

The watermark bit is 1

Else

The watermark bit is 0

End If

Step 6: Reconstruct the watermark using the extracted watermark bits.

End

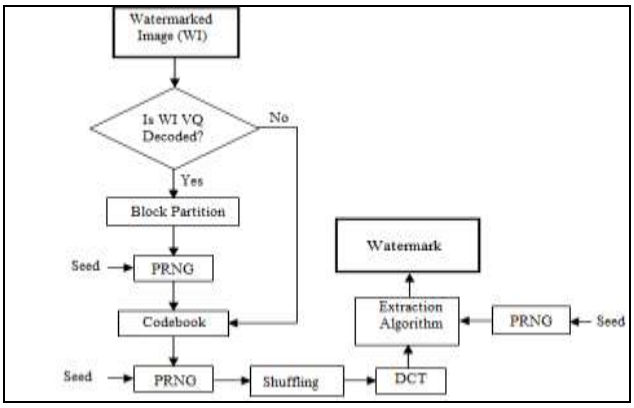


Figure 4: Block diagram of the watermark extraction process

4. EXPERIMENTAL RESULTS

In this section, the simulation results are presented to evaluate the performance of the proposed watermark scheme. We have carried out our proposed algorithm on a set of standard gray level images, but in this paper only the results of the most popular image 'Pepper' of size *512×512* with *256* gray levels is considered. Here we have taken a binary logo of size *32×32* as a watermark. Both images are shown in *Figure 5.*

(a) **(b)**

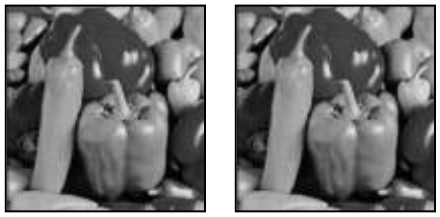
Figure 5: (a) The original Pepper Image (b) The watermark

In our experiments, we used modified LBG algorithm to train a codebook of size *1024* where the codeword size is *8×8*. In watermark embedding process we have taken the value of $\alpha=0.2$. In this paper, we compute the Peak signal to noise ratio (PSNR) value to evaluate the quality between the original image and watermarked image or the attacked watermark image.The PSNR is calculated as

$$PSNR = 10\log_{10}\left(\frac{255^2}{MSE}\right)db \quad (3)$$

Where $MSE = \frac{1}{N^2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\left(x(i,j) - x_w(i,j)\right)^2$ (4)

Figure 6 shows the VQ decoded original image and watermark image along their PSNR value.



(a) PSNR 30.9526 **(b) PSNR 30.3849**

Figure 6: VQ reconstructed (a) Original Pepper Image (b) Watermark Image

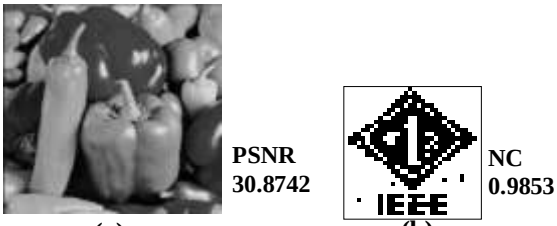
We have used the normalized cross correlation (NC) to evaluate the distortions in the extracted watermark by possible attacks. The value of NC lies between 0 and 1. The bigger NC value is the better watermark robustness. The NC value is evaluated as follow:

$$NC = \frac{\sum_{i=0}^{M-1}\sum_{j=0}^{M-1}\left(w(i,j)w'(i,j)\right)}{\sum_{i=0}^{M-1}\sum_{j=0}^{M-1}\left(w(i,j)\right)^2} \quad (5)$$

Where w and w' are the watermark and the modified watermark respectively. Each watermark is of size $M \times M$. In the following subsection, we will present the simulation results to show the effectiveness of our watermark algorithm. We have executed the following attacks on the watermark image to test the robustness of our proposed watermark scheme.

Image Filtering

In filtering operation, we apply Gaussian *3×3* low pass filter on watermark image as shown in *Figure 7(a).* The extracted watermark is shown in *Figure 7(b).*



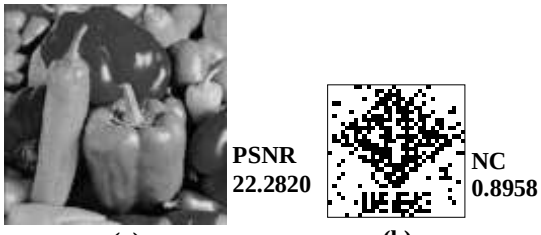
PSNR 30.8742 **NC 0.9853**

(a) **(b)**

Figure 7: (a) Watermark Image after applying Low pass Gaussian Filter (b) Extracted Watermark

Image Noising

We add Salt & Pepper noise of variance 0.01to the watermarked image. The presence of this noise reduces the PSNR value of the watermark image. However, the retrieved watermark is still recognizable as shown in *Figure 8*.



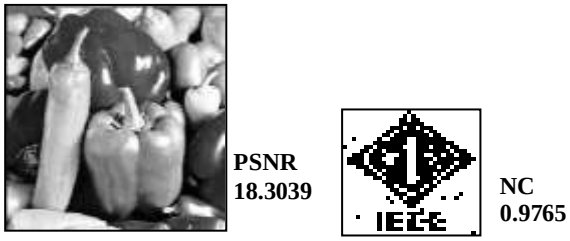
PSNR 22.2820 **NC 0.8958**

(a) **(b)**

Figure 8: (a) Watermark image after noising (b) Extracted Watermark

Histogram Equalization

We enhanced the watermark image by applying the histogram equalization as shown in Figure 9(a). The PSNR value of enhanced image is reduced to 18.3039 dB. However, the retrieved watermark is still recognizable as shown in *Figure 9(b).*



PSNR 18.3039 **NC 0.9765**

(a) **(b)**

Figure 9: (a) Watermark image after histogram Equalization (b) Extracted Watermark

Image Cropping

Several cropping operations like quarter-cropped and half-cropped are performed on the watermark images. In quarter-cropped attacks, the watermark image is cropped by one-fourth from the upper left corner where in half-cropped attack, one-half of the watermark image is replaced by some other image or pixel values. From the simulation results we find that cropping operation of quarter-croped has less influence on the visual quality of the extracted watermark. The quarter-cropped watermarked images are shown in *Figure 10*. The extracted watermarks are shown in bellow of the cropped images with their corresponding NC value.

NC 0.9848 **NC 0.9848** **NC 0.9075**

(a) Quarter-cropped and filled with pixel value 255 **(b) Quarter-cropped and filled with pixel value 0** **(c) Quarter-cropped replaced with the original image**

Figure 10: The quarter-cropped, watermark images

In half-cropped attack, the quality of extracted watermark is degraded in some cases but still they are recognizable as shown *Figure 11.*

(a) Half-cropped and replaced with pixel value 255

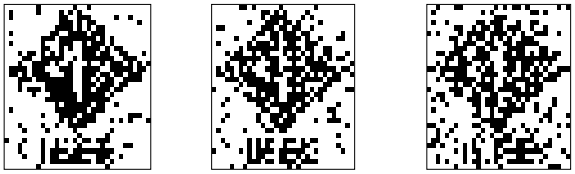
(b) Half-cropped and replaced with pixel value 0

(c) Half-cropped replaced with the original image

Figure 11: The half-cropped, watermark images

JPEG Image Compression

In general image is preferred to compress before transmission over the Internet. JPEG compression technique is generally used. Here JPEG compression is applied on the basis of different compression ratio to the watermarked image and the extracted watermark is shown in *Figure 12.*



(a)Compression ratio=12.6 NC=0.9019

(b)Compression ratio=13.2 NC=0.8905

(c)Compression ratio=14.9 NC=0.8503

Fig. 12: Extracted watermarks from the image Pepper

5. CONCLUSION

This paper proposes a hybrid watermarking scheme that compresses the images using VQ and embeds the logo in the codebook after DCT operation. At the sending-end, before embedding the logo, the position of codewords are shuffled using a pn-sequence for security reason and the same pn-sequence is used at the receiving-end to reconstruct the codebook and extract the logo. This strengthens the confidentiality of the watermarking as only the legal user, who knows the pn-sequence, can recover the logo or copyright information. Thus the proposed scheme supports both compressed transmission and the protection of the intellectual property over the open channel like Internet. The proposed watermarking scheme has been simulated over several images for logo extraction and image processing attacks such as filtering, noising, cropping, JPEG compression etc and satisfactory results have been obtained.

6. REFERENCES

[1] Chun-Shien Lu," Multimedia Security: Steganography and Digital Watermarking Techniques for Protection of Intellectual Property", Idea Group Publishing, 2005.

[2] P. Wolfgang, E.J. Delp, A watermark for digital images, Proc. IEEE International. Conf. on Image Processing (ICIP'96),Vol. III, Lausanne, Switzerland, pp. 219-222,September 1996,

[3] N. Nikolaidis, I. Pitas, "Robust image watermarking in the spatial domain", *Signal Processing*. Vol no.66 ,pp. 385-403,1998

[4] M. U. Celik, G. Sharma, A. M. Tekalp, E. Saber, "Lossless generalized-LSB data embedding", *IEEE Trans. Image Process.*, 14 (2005) 253-266.

[5] Mauro Barni, Franco Bartolini, Vito Cappellini and Alessandro Piva, "A DCT-domain system for robust image watermarking", *Signal Processing* 66 (1998) pp no. 357—372.

[6] Wai C. Chu, "DCT-based Image Watermarking using subsampling", *IEEE Transactions of Multimedia,* Vol. 5,No. 1,pp-34-38,March- 2003.

[7] Jiang-Lung Liu, Der-Chyuan Lou, Ming-Chang Chang and Hao-Kuan Tso, "A robust watermarking scheme using self-reference image", *Commuter Standards & Interfaces* 28(2006) 356-367.

[8] A. H. Taherinia, and M. Jamzad, "A Robust Image Watermarking using TwoLevel DCT and Wavelet Packets Denoising", *International Conference on Availability, Reliability and Security*, pp no 150-157,2009.

[9] Mei Jiansheng, Li Sukang and Tan Xiaomei, "A Digital Watermarking Algorithm Based On DCT and DWT", *International Symposium on Web Information Systems and Applications (WISA'09)*, pp no-104-107, 2009.

[10] Ali Al-Haj," Combined DWT-DCT Digital Image Watermarking", Journal of Computer Science 3 (9): 740-746, 2007.

[11] Rafael C. Gonzalez and Richard E. Woods, "Digital Image Processing", Pearson Edition, 2005.

[12] Hsien-Chu Wu and Chin-Chen Chang, "A novel digital image watermarking scheme based on the vector quantization technique", *Computers and Security*, Vol. 24, pp. 460-471, 2005.

[13] Cheng-Hsing Yang, Chi-Yao Weng, Shiuh-Jeng and Hung-Min Sun, "Grouping Strategies for Promoting Image Quality of Watermarking on the basis of Vector Quantization", *Journal of Visual Communication and Image Representation*, Vol 21,pp.39-55, 2010.

[14] H. C. Huang, F. H. Wang, and J. S. Pan, "Efficient and robust watermarking algorithm with vector quantization", Electronics Letters, 37 (2001) 826-828.

[15] Jau-JiShen, andJia-Min Ren," A robust associative watermarking technique based on vectorquantization", Digital Signal Processing (2009).

[16] A. Gersho and R. M. Gray, "Vector Quantization and Signal Compression", *Kluwer Academic Publishers, Boston, MA;* 1991.

[17] David Salomon, "Data Compression: The Complete Reference", *Springer international Edition*, 2005.

[18] Y. Linde, A. Buzo and R. M. Gray, "An algorithm for vector quantizer design", *IEEE Transactions on Communications*, vol. 28, pp. 84-95,1980.