Variable Dimension Vector Quantization Based Image Watermarking

S. Sethu Selvi
Department of ECE
Indian Institute of Science
Bangalore 560 012
selvi@protocol.ece.iisc.ernet.in

Anamitra Makur
Department of ECE
Indian Institute of Science
Bangalore 560 012
amakur@ece.iisc.ernet.in

Abstract

A watermarking method based on variable dimension vector quantization for hiding information in images is presented in this paper. Watermark bits are embedded in the dimension information of the variable dimension reconstruction blocks of the cover or input image. Watermark extraction for oblivious watermarking is carried out by identifying the dimension of the reconstruction block. A variation of the scheme for cover escrow watermarking is also presented to increase robustness. The simulation results prove the effectiveness of the proposed scheme and this scheme gives comparable capacity with the existing schemes.

1 Introduction

1.1 Watermarking

Data hiding represents a class of processes used to embed data, such as copyright information, into various forms of media such as image, audio or text with a minimum amount of degradation to the host signal. Data hiding in images is known as watermarking. A brief review describing a wide range of watermarking techniques is available in [1]. A watermark should be imperceptible, be difficult or impossible to remove, survive lossy compression like JPEG and a number of attacks (see [2]), and unambiguously identify the owner upon retrieval.

Figure 1 illustrates the watermark embedding and detection process. The output of the detection process is either the recovered watermark W or some kind of confidence measure indicating how likely it is for the given watermark at the input to be present in the image under inspection.

While visible watermarks are useful for conveying an immediate claim of ownership, invisible watermarks are useful in identifying the author, owner, distributor or authorized consumer of an image. Private watermarks (also referred to as non blind or cover escrow) require at least the original image during detection, and are more robust. Public watermarks (also referred to as blind or oblivious) require neither the original image nor the embedded watermark in extracting the information bits (the watermark) from the marked image.

Spatial domain image watermarking techniques include flipping the least significant bit of chosen pixels in an image [3]. This scheme is susceptible to any human or noisy modification. A more robust technique superimposes a watermark over an area of the image and then add some fixed intensity value for the watermark to the varied pixel values of the image [4]. However, image cropping can be used to eliminate the watermark.

Example of frequency domain watermarking is using DCT [5], where the values of chosen frequencies are altered from the original. This method is less susceptible to cropping, but has a tradeoff between invisibility and decodability. It also achieves compression. Similarly, [6] presents an approach to image watermarking utilizing fractal image compression, while [7] proposes a scheme based on Vector Quantization (VQ).

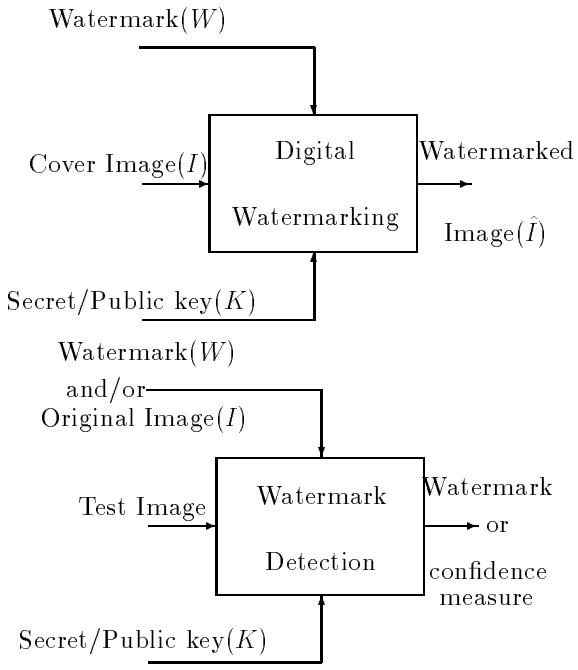

Figure 1: Digital Watermark Embedding & Detection

1.2 Vector Quantization

Vector quantization is well reviewed in [8]. It is a lossy block data compression technique wherein the vectors are quantized rather than scalars. The input image is partitioned into two dimensional blocks $\mathbf{x}$ of dimension k. Each input block is mapped to a finite set of codevectors $C = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_M\}$, which form the codebook, shared by both the encoder and the decoder. The index of the codevector which best matches the input block $\mathbf{x}$ according to some cost function is sent to the decoder, which uses the index to look up the reconstruction vector for $\mathbf{x}$ in its copy of the codebook. While conventional VQ is a fixed dimension scheme, a more general (hence better performing) variable dimension VQ (VDVQ) is possible.

1.2.1 Variable Dimension VQ

VDVQ has codevectors of varying dimensions. In this, active blocks are coded with low dimensional vectors to permit high rates, and inactive blocks are coded with higher dimensional vectors with low rates. A strategy for VDVQ is to choose the vector dimension depending on a performance measure like distortion, rate etc. The decision on the vector dimension and the coding is done simultaneously.

A review of various VDVQ encoding (for 1-D and 2-D sources) and codebook design with applications to speech and image coding is found in [9]. For example, Chou and Lookabaugh [10] propose a variable-to-variable length VQ for 1-D source. Dynamic programming is used for encoding using $J = D + \lambda R$ as the cost function to be minimized, where D is the distortion, R is the rate, and λ is the Lagrange multiplier. The optimal encoding is then found by backtracking and therefore this encoding has infinite delay. This delay is reduced to a finite expected delay by a trellis based sequential encoding in [11]. An adaptive online VQ with variable sized codevectors in the codebook is proposed in [12]. This algorithm is similar to dictionary based textual substitution algorithms.

1.2.2 VDVQ Encoding of Images

The algorithm used by us is based on tree encoding, as presented in [13]. Given a codebook containing codevectors of varying dimensions belonging to the set $K = \{k_1, k_2, \ldots, k_N\}$, optimal VDVQ encoding achieves partitioning of the blocks of the input image together with the matching of the input blocks to codevectors, such as to minimize a cost function J. The encoding algorithm uses delay decision encoding or Viterbi algorithm.

The encoding is started from the upper left corner of the image. The root node at depth $d = 0$ is anchored at this starting point. The minimum cost codevector of dimension k_1 is found for the image block of the same dimension. The dimension k_1 implies any two dimensional block of size $X \times Y$. This corresponds to the first child l_1 of the root node. The encoding cost and the index of the best matched codevector are recorded for this child. The above process is repeated for other dimensions $k_2, k_3, \ldots, k_N$ and N children $l_2, l_3, \ldots, l_N$ at depth $d = 1$ are formed.

The next point of encoding is selected by the *Raster Scan* and $k_1, k_2, \ldots, k_N$ dimensional blocks are formed from that point and the N children are created with the accumulated cost and index. This tree growing is continued till the entire image is encoded. At this stage, we have all possible encodings of the image with different dimensional codevectors. The path having the minimum accumulated cost can be declared and the resulting encoding is the optimal encoding.

Raster Scan decides only the starting point of the new block. It may result in a new block with some pixels already encoded. In this case, we use *First* coverage such that the cost is computed only for the pixels yet to be encoded. While decoding, the first codevector that encoded some pixel defines the decoded image.

To reduce the exponentially increasing size of the tree, a suboptimal algorithm called M algorithm is employed. In this, the search is narrowed to at most M least cost paths at any depth by pruning the tree.

1.3 Watermarking Based on VQ

A digital image watermarking technique based on VQ is presented in [7]. The codebook is partitioned into a few clusters of close codevectors, such that any pair of codevectors in a cluster have Euclidean distance below certain threshold. Let the best match codevector for an input block fall in some cluster with a cluster index i, and let the cluster have size $2^{n(i)}$. An $n(i)$ bit integer g, the watermark information, is embedded by transmitting an index corresponding to the $(j+g) \bmod 2^{n(i)}$-th codevector in this cluster. The scheme is cover escrow since the extraction requires the original image. The codebook partition is used as the secret key. A tampering is detected if the received index and the best match index do not belong to the same cluster. For 512×512 Lenna image and input block size of 4×4, this scheme reports an extra distortion of 2.79 msepp introduced due to watermarking.

In this paper, we propose both private and public watermarking techniques based on VDVQ where the dimension of a block carries the watermark information. In Section 2, we describe the proposed watermarking schemes. Simulation results and conclusions are provided in Section 3.

2 Proposed Watermarking Scheme

We consider the watermark information to be a bit sequence $\{b_i\}$, $b_i \in \{0, 1\}$, and two dimensions in our codebook ($N = 2$, $K = \{k_1, k_2\}$). In general, n bits can be embedded at a time if $N \geq 2^n$. Given a threshold T, for each input block $\mathbf{x}$, the embedding process is performed block by block as follows.

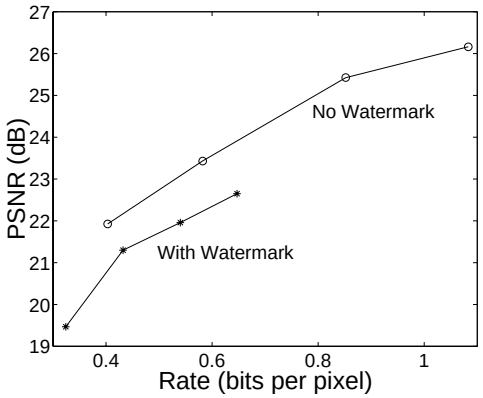

Figure 2: Rate vs PSNR for Lenna

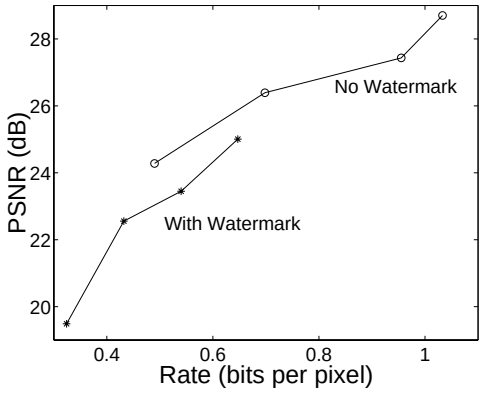

Figure 3: Rate vs PSNR for Peppers

1. Find the best matching codevector $\mathbf{c}_{i,m}$ of dimension k_m, $m \in \{1, 2\}$, for the input block $\mathbf{x}$. Let the cost of encoding $\mathbf{x}$ with $\mathbf{c}_{i,m}$ be J_m.
2. If $|J_1 - J_2| < T$, the dimension of the block is selected according to b_i. If $b_i = 0$, then $\mathbf{x}$ is encoded as $\mathbf{c}_{i,1}$, and if $b_i = 1$, then $\mathbf{x}$ is encoded as $\mathbf{c}_{i,2}$.
3. If $|J_1 - J_2| > T$, no watermark bit is embedded in that block and $\mathbf{x}$ is encoded as $\mathbf{c}_{i,1}$ or $\mathbf{c}_{i,2}$, depending upon which codevector has the least cost.

Encoding details such as the starting point, *Raster Scan*, and the cost are as given in Section 1.2.2. The cost J is calculated per pixel as two different sized blocks are compared, and only for the pixels that are not encoded (*First* coverage) and for the pixels that lie inside the image boundaries. The number of watermark bits embedded depends upon the cover image used for watermarking. The threshold T acts as the secret key for extraction of the watermark bits. Depending upon the threshold T, both public and private watermarking schemes can be described. Since the watermark bits reside in the codevector indices, the codebook acts as a secret key against standard known attacks. Image based attacks like compression or geometrical transformation are not applicable directly since the image here is compressed.

2.1 Oblivious Watermarking

Blind/oblivious watermarking schemes are used to detect copyright violations of images on the Internet. This remains the most challenging problem since it requires neither the original image nor the embedded watermark.

If the threshold $T = \infty$, then the proposed watermarking becomes a blind scheme. If $T = \infty$, $|J_1 - J_2|$ is always less than ∞ and each block has a watermark bit embedded into it. The number of watermark bits is equal to the number of VDVQ blocks used for encoding the image. These bits can be extracted by merely knowing the dimension of the block, which is implied by the codevector index transmitted by the encoder.

If the dimension of the codevector corresponding to the index received from the encoder is k_m, then the watermark bit embedded is $b_i = 0$ for $m = 1$, and $b_i = 1$ otherwise. This scheme is used for watermark extraction only if the user knows that there are watermark bits in the compressed image. Otherwise, it is decoded as a mere compressed image without any watermark information.

2.2 Cover Escrow Watermarking

Blind watermarking algorithms can also usually be used in cover escrow/non blind watermarking, thus increasing robustness. The main applications of these schemes are to prove ownership and copy control in DVD etc. where the user is allowed or not allowed to play the contents. The original image is used as a hint to find where the watermark information is.

If the threshold $T < \infty$, then the watermark extraction becomes a non blind scheme. If $T = 0$, then no watermark bits are embedded. For $0 < T < \infty$, the number of watermark bits increases with increasing threshold. The watermark extraction process is as follows.

The owner knows T and also has the original image. So, as in encoding, the best codevectors $\mathbf{c}_{i,1}$ and $\mathbf{c}_{i,2}$ are found for each input block of the original image. If the corresponding encoding costs J_1 and J_2 satisfy $|J_1 - J_2| < T$, then the owner knows that a watermark bit is embedded in this block. If the dimension of the codevector corresponding to the received index is k_m, then the embedded watermark bit is $b_i = 0$ if $m = 1$ and $b_i = 1$ if $m = 2$. The hacker does not have the original image to compute the encoding costs J_1 and J_2 even if he has the codebook and also the threshold T. So, the watermark information cannot be extracted without the original image.

3 Simulation Results and Conclusion

The proposed watermark scheme was simulated for various images. The performance measures used to

Image	Codebook dimensions	PSNR Loss
Lenna	2 , 3	2 dB
	2 , 4	1 dB
	3 , 4	1.5 dB
Peppers	2 , 3	4 dB
	2 , 4	0.8 dB
	3 , 4	2.5 dB

Table 1: PSNR loss in dB for different codebook dimensions at a typical rate of 0.65bpp

Image	PSNR Loss
Baboon	0.5 dB
Peppers	0.5 dB
Parrot	0.5 dB
Barbara	1 dB
Lenna	1 dB
Boats	2 dB

Table 2: PSNR loss in dB for different images at typical rate of 0.7bpp

evaluate the proposed scheme are the Peak Signal to Noise Ratio (PSNR), Rate in bits per pixel, and Capacity (the number of bits embedded in the input image) in bits per pixel.

The codebooks used for encoding were designed using the LBG algorithm (popular VQ codebook design algorithm, see [8]) for a large training set of different images. The initial codebook of required size was designed by selecting random blocks of required dimensions from the training set. VDVQ encoding for images was used and the centroid was calculated as in LBG algorithm. The cost function used was distortion only ($\lambda = 0$). The codebooks designed were of sizes 8, 16, 32 & 64 and had codevectors of dimensions $\{2,3\}$, $\{2,4\}$ and $\{3,4\}$. The watermark bits were generated randomly such that both 0 and 1 are equiprobable. The same bit pattern was used for all the simulations.

3.1 Results for Oblivious Watermarking

The Rate vs PSNR plots without and with oblivious watermarking are shown in Figures 2 and 3 for Lenna and Peppers for codevector dimensions $\{2,4\}$. Rate is varied by varying the codebook size from 8 to 64.

The rate-distortion performance of the image with watermark is inferior to that without watermark. This is expected since, in some cases, some other dimensional codevectors than the best match are forced depending on the watermarking bits. While this gives rise to a decrease in PSNR, some large dimensional codevectors are forced in watermarking compared to encoding without watermarking, resulting also in the rate being less than that without watermarking. Capacity obtained for both images is 0.108bpp. Table 1 shows the PSNR loss (reduction in PSNR for watermarking compared to without watermarking) for

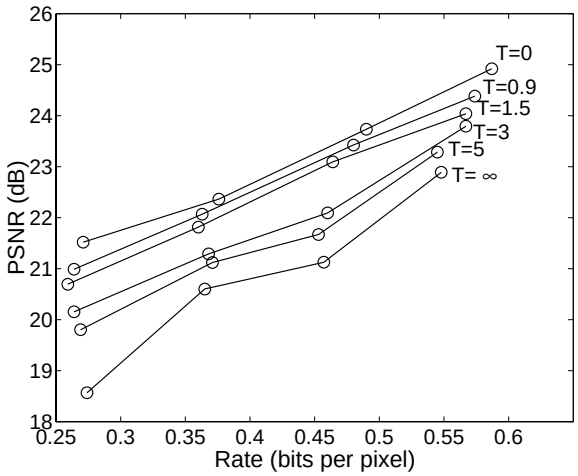

Figure 4: Rate vs PSNR for Lenna

Lenna and Peppers for different codevector dimensions at a typical rate of 0.65bpp. It is observed that codevector dimensions $\{2,4\}$ give the least PSNR loss for both the images.

The simulations were also done on a number of images for different codebook sizes and codevector dimensions. Table 2 reports the PSNR loss at a typical rate of 0.7bpp for different images for codevector dimensions of $\{2,4\}$. It is seen clearly that some images give much less PSNR loss compared to some other images, hence the distortion penalty is image specific. It is seen that there is very little distortion penalty due to this watermarking scheme.

3.2 Results for Cover Escrow Watermarking

Simulation results for cover escrow scheme, presented in this section, are for a 64×64 part of Lenna, and for T varied from 0 to ∞. The reported threshold T is normalized with respect to the distortion obtained without watermarking for that particular image.

Figure 4 shows the variation of Rate vs PSNR for different normalized thresholds T. Rate is varied by varying the codebook sizes from 8 to 64 and the codevector dimensions used are $\{3,4\}$. As the threshold increases, PSNR decreases due to increase in capacity. For a codebook size of 64, as the capacity increases from 0bpp to 0.09bpp for a threshold range of $0 - \infty$, the PSNR drops from 24.9dB to 22.8dB, which means a maximum PSNR loss of 2.1dB.

The variation of PSNR loss vs T for different codevector dimensions is shown in Figure 5 for a typical rate of 0.5bpp. Similar to the blind scheme, it is seen that for codevector dimensions $\{2,4\}$ the PSNR loss is minimal compared to other codevector dimensions. Further, the PSNR loss does not change over a wide range of T. The maximum PSNR loss obtained is 2.4dB.

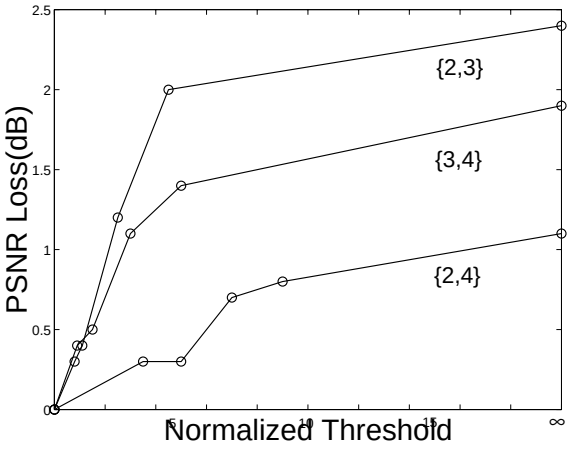

Figure 5: Normalized threshold vs PSNR loss in dB at a typical rate of 0.5bpp

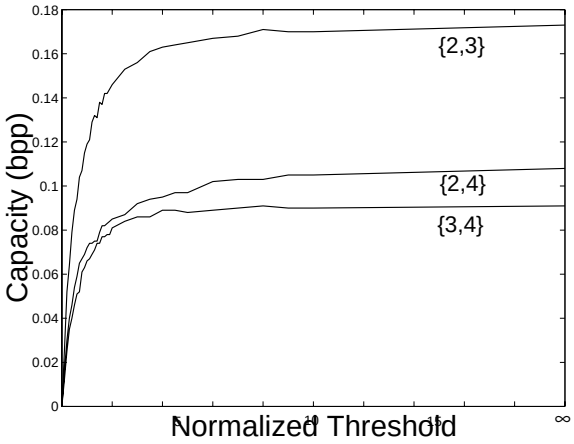

Figure 6: Normalized threshold vs Capacity

The variation of capacity with T is depicted in Figure 6 for codebook size 64 and for different codevector dimensions. The capacity is higher for smaller codevector dimensions, due to the increase in the number of input blocks for smaller codevector dimensions. Also, the range of variation of capacity with respect to the threshold is very small. For codevector dimensions $\{2, 4\}$, the capacity varies from 0bpp to 0.108bpp as the threshold varies from 0 to ∞.

The original 128×128 part of Lenna, the reconstructed image without watermarking, and the one with blind watermarking are shown in Figure 7, Figure 8 and Figure 9 respectively. These images are for a codebook size of 64 and codevector dimensions of $\{2, 4\}$, and has a capacity of 0.1066bpp. The number of encoded blocks decreases from 2967 for the image without watermark to 1746 for the watermarked image, reflecting a decrease in rate. This rate reduction is responsible for the PSNR reduction of about 4dB

Figure 7: Original 128×128 image

Figure 8: No Watermark
PSNR = 28.3dB Rate = 1.09

observed in this case.

Comparing with the existing methods, the proposed scheme gives comparable or better capacity. In [14], the transform based blind watermarking method gives a capacity of 0.03bpp and the edge based blind watermarking method gives a capacity of 0.006bpp. In [7], the VQ based watermarking reports a capacity of 0.127bpp. In comparison, the proposed blind watermarking scheme gives a capacity of 0.108bpp.

3.3 Variations of the proposed scheme

The dimensions allowed for encoding in the proposed scheme are not anyway constrained other than dependence on the watermark bits. We can have the following variations of the above algorithm by imposing some constraints on the dimensions allowed for a block. These constraints are relaxed if the required dimensional block is not available in the codebook. The watermark bits are embedded only in those blocks where both dimensions are possible.

(i) No overlap pixels : The dimension of the blocks are constrained such that there are no overlapping pixels. Pixels are allowed to extend beyond image boundaries.

Figure 9: With Watermark
PSNR = 24.2dB Rate = 0.64

(ii) No outside pixels : The dimension of the blocks are constrained such that the pixels are not allowed to extend beyond image boundaries. Overlapping of pixels are allowed.

(iii) No overlap & outside pixels : Constraints of both (i) and (ii).

Due to the constraints, the capacity of these schemes are reduced. For example, for a codebook size of 64 and codevector dimensions $\{2, 4\}$, unconstrained, constraint (i), (ii) and (iii) respectively gives a capacity of 0.108, 0.094, 0.101 and 0.088 bpp. Other codevector dimensions give still more reduction, such as a 50% reduction in capacity (0.04bpp from the unconstrained capacity of 0.09bpp) for constraint (i) with a codebook size of 64 and codevector dimensions $\{3, 4\}$.

3.4 Conclusion

A new method for digital watermarking for images has been proposed in this paper. The proposed scheme produces watermarks on VDVQ compressed images that are not detectable by visual inspection. The scheme is secret, efficient, and robust. Also, the watermark is inexpensive to create, detect, and verify. Both oblivious and cover escrow versions of the proposed watermarking algorithm are explored. Some variations of the proposed scheme are also mentioned. Simulations were done on various images by designing different codebooks of different codevector dimensions. Experimental results are reported to prove the effectiveness of the proposed algorithm.

References

[1] F. A. Petitcolas, R. J. Anderson and M. G. Kuhn "Information Hiding - A Survey" *Proceedings of IEEE* vol. 87, pp. 1062-1078, July 1999.

[2] M. Kutter and F. A. Petitcolas "A Fair Benchmark for Image Watermarking Systems" *Proc. SPIE Security and Watermarking of Multimedia Contents* vol. 3657, pp. 226-239, January 1999.

[3] R. G. Van Schyndel, A. Z. Tirkel, N. Mee and C. F. Osborne "A Digital Watermark" *Proceedings of ICIP* vol. 2, pp. 86-90, November 1994.

[4] M. D. Swanson, B. Zhu and A. H. Tewfik "Transparent Robust Image Watermarking" *Proceedings of ICIP* vol. 3, pp. 211-214, September 1996.

[5] I. Cox, J. Kilian, F. T. Leighton and T. Shamoon "Secure Spread Spectrum Watermarking for Multimedia" *IEEE Transactions on Image Processing* vol. 6, No. 12, pp. 1673-1687, December 1997.

[6] P. Davern and M. Scott "Fractal Based Image Steganography" *First International Workshop on Information Hiding* Lecture Notes in Computer Science, pp. 279-294, 1996.

[7] Z. M. Lu and S. H. Sun "Digital Image Watermarking Technique Based on Vector Quantization" *IEEE Electronics Letters* vol. 36, No. 4, pp. 303-305, February 2000.

[8] R. M. Gray "Vector Quantization" *IEEE ASSP Magazine* vol. 1, pp. 4-29, April 1984.

[9] S. Sethu Selvi and Anamitra Makur "A Review of Variable Dimension Vector Quantizers & their Applications to Speech and Image Coding" *Journal of Electro Technology* vol. 41, No. 3 & 4, pp. 38-70, September - December 1997.

[10] P. A. Chou and T. Lookabaugh "Variable Dimension VQ of LPC of Speech" *Proceedings of ICASSP* vol. 1, pp. 505-508, 1994.

[11] Anamitra Makur and K. P. Subbalakshmi "Variable Dimension VQ Encoding and Codebook Design" *IEEE Transactions on Communications* vol. 45, pp. 897-899, August 1997.

[12] C. Constantinescu and J. A. Storer "Online Adaptive VQ with Variable Size Codebook Entries" *Proceedings of DCC* pp. 32-41, 1993.

[13] S. Sethu Selvi and Anamitra Makur "Variable Dimension Vector Quantization of Images by Two Dimensional Multipath Search" *Proceedings of ICVGIP* pp. 40-48, December 1998.

[14] Chauhan H. Nileshkumar "Transform Based Blind Steganography of Compressed Images" *M. E. Project Report* Department of ECE, IISc, January 2000.