

A Technique for Image Data Hiding and Reconstruction without Host Image

J. J. Chae and B. S. Manjunath

Department of Electrical and Computer Engineering
University of California, Santa Barbara, CA 93106
chaejj, manj@iplab.ece.ucsb.edu

ABSTRACT

A new technique for embedding image data that can be recovered in the absence of the original host image, is presented. The data to be embedded, referred to as the signature data, is inserted into the host image in the DCT domain. The signature DCT coefficients are encoded using a lattice coding scheme before embedding. Each block of host DCT coefficients is first checked for its texture content and the signed codes are appropriately inserted depending on a local texture measure. Experimental results indicate that high quality embedding is possible, with no visible distortions. Signature images can be recovered even when the embedded data is subject to significant lossy JPEG compression.

1 INTRODUCTION

Embedding signature images into other images and video has applications in data hiding and digital watermarking. In the past few years, much progress has been made in developing watermarking techniques that are robust to signal processing operations such as data compression^{1,2}. Signature information for watermarked images range from pseudorandom sequences to small image icons that can be easily recovered and authenticated. Emphasis is on authentication rather than the quantity and quality of the recovered signature.

In contrast, in perceptual data hiding, one is interested in embedding and recovering high quality multimedia data, such as images, video and audio. The host multimedia data itself could be subject to signal processing operations, typically compression. Depending on the end user application, both lossy and lossless data embedding is of interest. Like in digital watermarking, two scenarios are possible: (a) the original host into which the data is embedded, is available. Some of the recent works address this problem^{1,3,4,5,6}. (b) the original host information is not available^{3,7,8}. This is a much more difficult problem.

In³ a spatial domain embedding based on bit replacement, is presented. Spatial domain strategies are quite sensitive to transformations on the embedded signal. In⁷ a robust technique to hide speech and video in compressed video is presented. The bit rate is about 2 bits per 8x8 block. A technique based on multidimensional lattice coding for data hiding is described in⁸ where the data rate is slightly less than 1% of the host data.

This paper addresses specifically the problem of image data hiding and recovery in the absence of host image data. Compared to related work, the proposed technique can embed significantly larger amount of signature data into the host - upto 25% of the host data, with little or no perceptual distortion.

A schematic of our data embedding scheme is shown in Figure 1. A key component of this scheme is embedding using multidimensional lattices (Section 2). Signature and host images are transformed using the block Discrete Cosine Transform (DCT). The block size chosen is 8 x 8 pixels. The signature coefficients are quantized in two steps. First, by using the standard JPEG quantization matrix, and then by a user specified signature quantization matrix. The signature quantization matrix determines the relative size of signature data compare to the host data, thus controlling

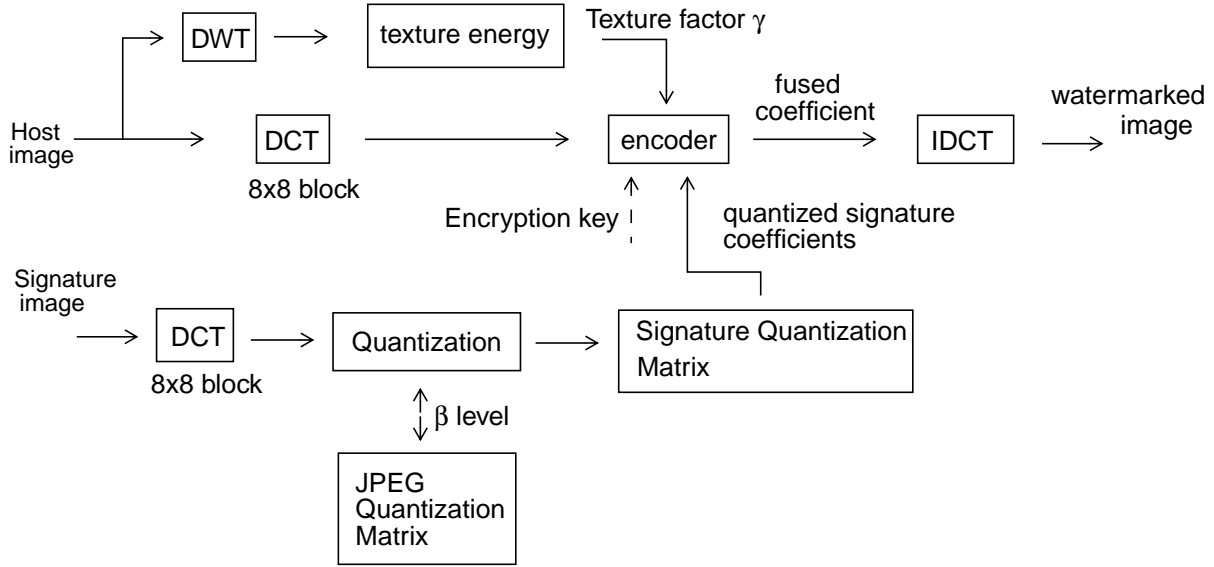


FIGURE 1. Aschematic of data embedding. Data is embedded in the bolck DCT domain. Signature DCT coefficients are quantized, coded using lattice codes and adaptively embedded into the host DCT coefficients using a texture masking strategy. See Figure 5 for details of the encoder.

the quantity and quality of the embedded data (Section 3). These quantized signature coefficients are then encoded using the multidimensional lattices and inserted into the host DCT coefficients. This insertion is adaptive to the local texture content of the host image blocks and controlled by the block texture factor γ (Section 4). The steps in embedding are summarized in Section 5 and Section 6 concludes with experimental results.

2 EMBEDDING USING MULTIDIMENSIONAL LATTICES

2.1 Methodology

If the original host image is available, the operations of data injection and retrieval are, in fact, very similar to the channel coding and decoding operations in a typical digital communication system. Channel coding refers to the gamut of signal processing done before transmission of data over a noisy channel. In watermarking in the transform domain, the original host data is transformed, and the transformed coefficients are perturbed by a small amount in one of several possible ways in order to represent the signature data. When the watermarked image is compressed or modified by other image processing operations, noise is added to the already perturbed coefficients. The retrieval operation subtracts the received coefficients from the original ones to obtain the noisy perturbations. The true perturbations that represent the injected data are then estimated from the noisy data as best as possible.

In this work, we adopt a vector-based approach to hidden data injection^{4,9,10}. We group N transform coefficients to form an N -dimensional vector, and modify it by codes that represent the data to be embedded. The motivation for using vector perturbations as opposed to scalar perturbations follows from the realization that higher dimensional constellations usually result in lower probability of error for the same rate of data injection and the same noise statistics. In Figure 2, ‘ x ’ represents a host vector in an N -dimensional space. To embed data from an β -ary source with symbols $\{s_1, s_2, \dots, s_\beta\}$, we perturb the original vector so that the perturbation coincides with one of β corresponding channel codes. The perturbed vector is denoted by one of the ‘ o ’s in Figure 2, depending on the particular source symbol it represents. After the watermarked image has undergone compression or other transformations, a perturbed vector representing, for example symbol s_i in the diagram, may be received as a noisy vector ‘*’, as shown in Figure 2 (b). It is then an estimation problem to extract the transmitted symbol from the vector received. Assuming an additive

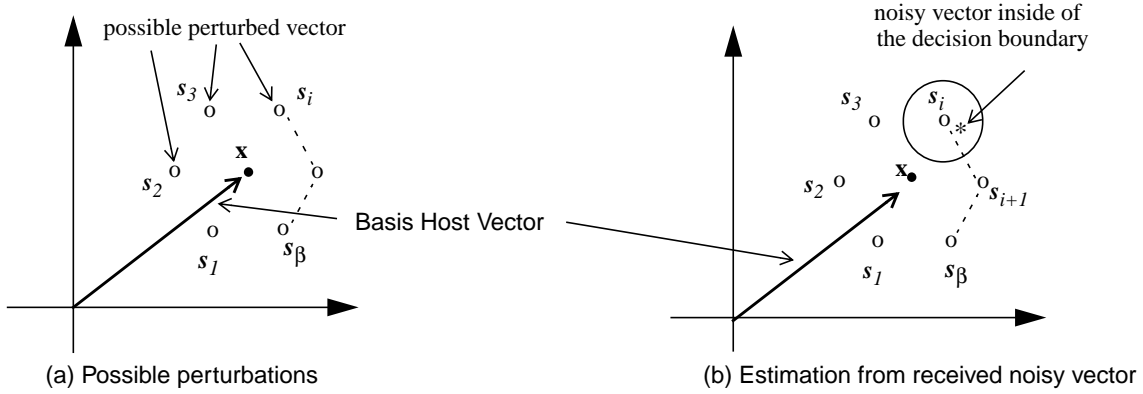


FIGURE 2: Encoding using lattice structures. (a) Possible β -ary perturbations of the host vector. (b) Possible noisy vector positions of original perturbed vector s_i after transformation.

Gaussian noise model, the received vector is decoded as representing the symbol, whose channel code it is closest to, in Euclidean distance.

Codes derived as subsets of multidimensional lattices have been shown to be very efficient for channel coding. In the following, we provide a brief review of lattice structures.

2.2 Lattice Structures

The Voronoi regions of various n -dimensional lattices can be used to construct n -dimensional quantizer cells for uniformly distributed inputs. It has been shown by Conway and Sloane ⁹ that some of these lattices produce very good channel codes, and yield high values of nominal coding gain. That is, for the same power constraint on the channel, the channel codes are maximally separated from each other so that they are most robust to noise. The lattices considered here are the root lattices and their duals, namely A_n , A_n^* , D_n , D_n^* , E_6 , E_8 , etc. If a_1, \dots, a_n are n linearly independent vectors in an m -dimensional Euclidean space with $m \geq n$, the set of all vectors

$$\mathbf{x} = u_1 a_1 + \dots + u_n a_n \quad (1)$$

where u_1, \dots, u_n are arbitrary integers, constitute an n -dimensional *root lattice* Λ_n ⁹. Further, if Λ is a lattice in \mathfrak{R}^n , the dual lattice Λ^* ⁹, consists of all points \mathbf{x} in the span of Λ such that $\mathbf{x} \cdot \mathbf{y} \in \mathbb{Z}$ for all $\mathbf{y} \in \Lambda$. Some common lattices and definitions are presented below.

For $n \geq 1$, A_n is the n -dimensional lattice consisting of the points (x_0, x_1, \dots, x_n) in \mathbb{Z}^{n+1} with $\sum x_i = 0$.

For $n \geq 2$, D_n consists of the points (x_1, x_2, \dots, x_n) in \mathbb{Z}^n with $\sum x_i$ even. In other words, if we color the integer lattice points alternately red and blue in a checkerboard coloring, D_n consists of the red points. In 4 dimensions the D_4 lattice is known to yield the best coding gain.

E_6 , E_8 , and Λ_{16} lattices give very good channel coding gains in 6, 8, and 16 dimensions respectively. E_8 is derived from the D_8 lattice, and is defined as the union of D_8 and the coset

TABLE 1: Code types and structure of the D_4 lattices

Shell No.	Squared Norm	Source codes	Number of codes
1	2	$(\pm 1, \pm 1, 0, 0)^P$	24
2	4	$(\pm 2, 0, 0, 0)^P,$ $(\pm 1, \pm 1, \pm 1, \pm 1)^P$	24
3	6	$(\pm 2, \pm 1, \pm 1, 0)^P$	96
4	8	$(\pm 2, \pm 2, 0, 0)^P$	24
5	10	$(\pm 2, \pm 2, \pm 1, \pm 1)^P,$ $(\pm 3, \pm 1, 0, 0)^P$	144

$$\left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right) + D_8.$$

In other words E_8 consists of the points (x_1, \dots, x_8) with $x_i \in Z$ and $\sum x_i$ even, together with the points (y_1, \dots, y_8) with $y_i \in Z + 1/2$ and $\sum y_i$ even. E_6 is a subspace of dimension 6 in E_8 , consisting of the points $(u_0, u_1, \dots, u_n) \in E_8$ with $u_6 = u_7 = -u_8$.

For a n -dimensional lattice Λ , the Voronoi region around any lattice point is the set of points in \mathfrak{R}^n closest to the lattice point. Therefore, the Voronoi region $V(0)$ around the origin is given as:

$$V(0) = \{ \mathbf{x} \in \mathfrak{R}^n \mid \|\mathbf{x}\| \leq \|\mathbf{x} - \mathbf{u}\| \text{ (for all nonzero } \mathbf{u} \in \Lambda) \} \quad (2)$$

2.3 The D_4 Lattice

It has been shown in ⁹ that some lattices produce very good spherical codes for channel coding. That is, for the same constraint on deviation from the true coefficient values, the channel codes are maximally separated from each other so that they are most robust to noise.

In general the D_4 root lattice produces the best channel code in 4 dimensions. For small noise, this lattice gives a nominal channel coding gain of 1.414 over binary encoding ⁹. As mentioned earlier, the D_4 lattice consists of the points (x_1, \dots, x_4) having integer coordinates with an even sum.

As in all lattices, the lattice points of the D_4 lattice fall on concentric shells of increasing distance from the all zero vector. For example, the 24 lattice points given by all permutations of $(\pm 1, \pm 1, 0, 0)$ lie on the first shell of the lattice at a distance $\sqrt{2}$ from the center. The second shell at distance 2 from the center contains 24 lattice points again, 8 of which are of type $(\pm 2, 0, 0, 0)$, and 16 are of type $(\pm 1, \pm 1, \pm 1, \pm 1)$. Table 1 shows the shell number, the squared norm, the lattice point types, and the number of lattice points for the first few shells of the D_4 lattice. The

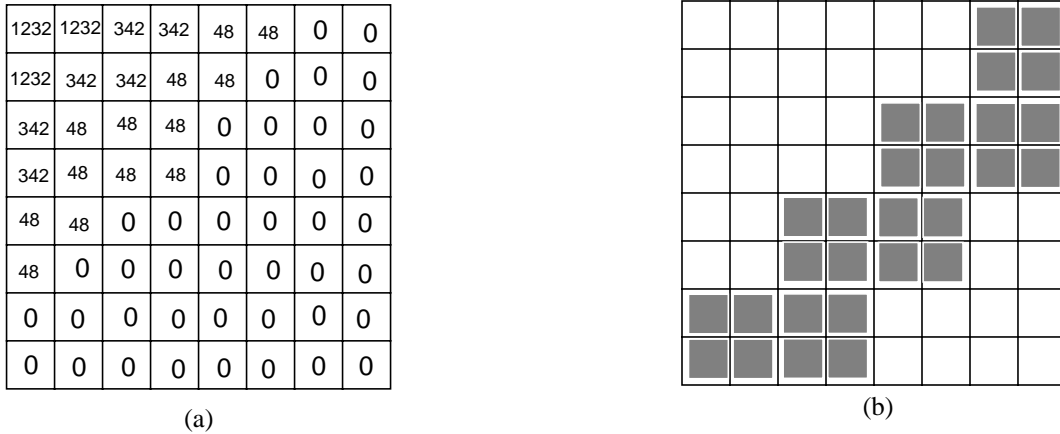


FIGURE 3. (a) Example of a signature quantization matrix for an 8 x 8 DCT coefficient block. This requires 112 host image coefficients to encode (see text for details). (b) A partitioning of the host DCT block for signal insertion (shaded regions). 28 coefficients are used in each block. Thus, four host DCT blocks ($4 \times 28 = 112$) are needed to embed one 8x8 signature DCT block.

superscript ‘p’ after the points in the table denote ‘all permutations of’ the elements constituting it. By choosing appropriate subsets of points from the lattice the rate for data embedding can be varied.

3 SIGNATURE IMAGE QUANTIZATION

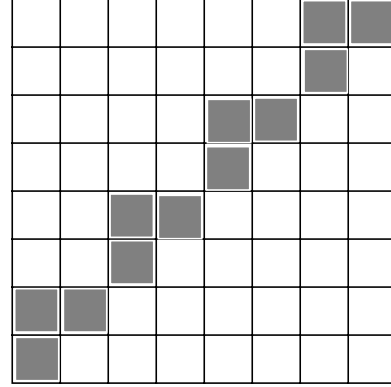
There is clearly a trade-off between data embedding quantity and quality of reconstruction. We propose a simple scheme here for quantizing signature image data using the block DCT quantization matrix. This approach enables, as demonstrated later in the experimental results, robust recovery of signature data when the embedded image is subject to JPEG compression.

Consider an 8 x 8 DCT coefficient matrix. From image compression and information theory, it is well known that low frequency coefficients require more bits than the high frequency ones. One such quantization matrix indicating the number of quantization levels for each of the 64 coefficients is shown in Figure 3(a). These quantized coefficients are embedded in a lattice structure as described in the previous section. For simplicity, we will consider only those shells in the lattice structure whose elements are $\{\pm 1, 0\}$. One way of distributing these coefficients is as follows:

- **Quantization Level=1232.** Use Lattice type E_8 : The first and second shells of E_8 lattice combined have 2400 code words, however, we use here 1232 code words from the combination of first shell and part of second shell in this lattice. Since an E_8 code has eight components, it requires 8 host coefficients to embed one E_8 code. There are 3 coefficients with this quantization, requiring 24 host coefficients to embed.
- **Quantization Level=342.** Use Lattice type E_6 : The first and second shells of E_6 lattice contains 342 code words. Six host coefficients are needed to embed an E_6 code. The six coefficients in the DCT matrix thus need 36 host image coefficients to embed.
- **Quantization Level =48.** Use Lattice type D_4 : The first two shells of D_4 are used to encode 48 levels. Each D_4 code requires four host coefficients. There are thirteen coefficients with this quantization, thus requiring 52 host coefficients.

1232	1232	1232	300	300	300	48	48
1232	1232	300	300	300	48	48	0
1232	300	300	300	48	48	0	0
300	300	300	48	48	0	0	0
300	300	48	48	0	0	0	0
300	48	48	0	0	0	0	0
300	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(a) signature quantization



(b) Selected positions for embedding

FIGURE 4. Another example of a signature quantization matrix and a corresponding host coefficient allocation. This requires 192 host coefficients, which are distributed over 16 blocks, 12 coefficients per block, as shown by the shaded regions in (b).

The scheme outlined above thus needs a total of 112 host coefficients to embed the 64 DCT coefficients from the signature image.

3.1 Host Coefficients

The next step in embedding is to identify the host coefficients which are affected by the data embedding procedure. The low frequency components contain most of the host signal energy but they can not be easily modified as such changes may become visible. The high frequency components, which usually pack the least amount of energy, could be easlily removed because of signal processing operations. This leaves us with the mid frequency components.

Consider an 8 x 8 block of host image coefficients, as shown in Figure 3(b). The shaded regions indicate the frequency components that are identified for encoding the signature image data. In this example, 28 host coefficients are used in each block, thus requiring four host DCT blocks to encode one signature block of Figure 3.

Another example of signature image quantization and the corresponding host coefficient allocation are shown in Figure 4. Notice that 192 host coefficients are needed for this case (6X for E_8 , 16X for E_6 , and 12X for $D_4 = 6 \times 8 + 16 \times 6 + 12 \times 4 = 192$). One possible way of distributing this is shown in Figure 4(b) where 12 host coefficients are identified for insertion. This requires a total of 16 host DCT blocks per signature block.

4 TEXTURE MASKING

The signature coefficients are adaptively embedded into the host image coefficients. Recall that insertions into host image regions with low texture information would result in visible distortions. The texture block factor γ controls the weighting of the signature coefficients for each 8 x 8 DCT host image block. We use a normalized measure of texture energy, defined as:

$$\mu_T(B) = \frac{\mu_D(B)}{\mu_W(B)} \quad (3)$$

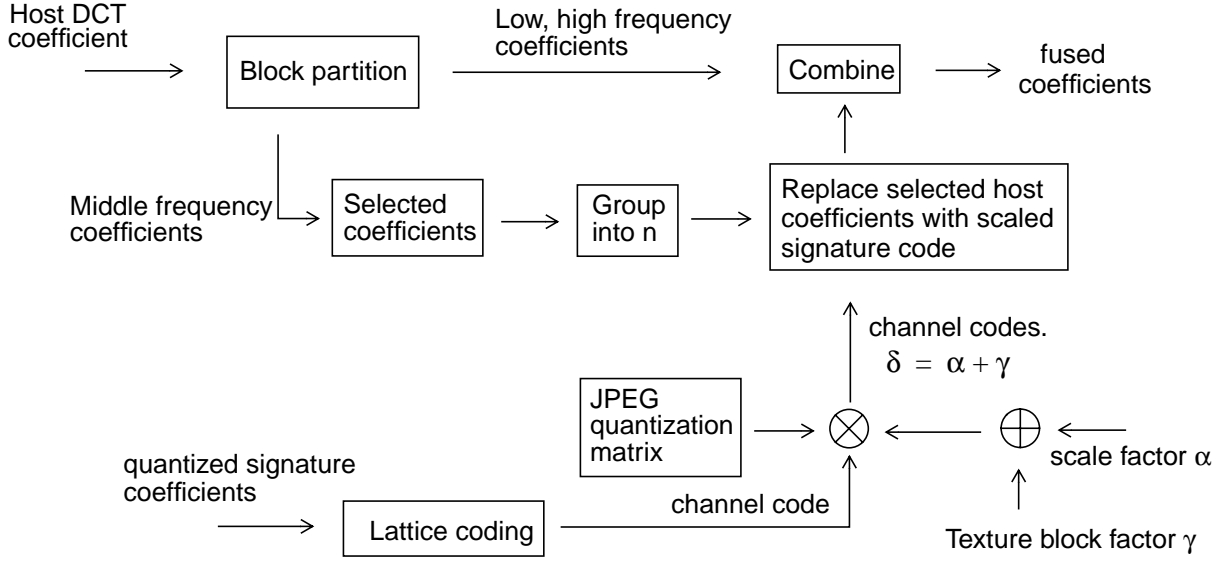


FIGURE 5. A schematic of the encoder

where $\mu_W(B)$ is the average energy in band B ($B = \{LH, HL, HH\}$) after a one level discrete wavelet decomposition of the host image and $\mu_D(B)$ is the average energy in band B of a given 8×8 host image block. $\mu_T(B)$ characterizes the given block texture energy for a given band B . A Haar wavelet transform is used in the experiments. If $\mu_T(B)$ exceeds a given threshold, say $T_H(B)$, then the corresponding block is considered to have significant texture in band B . If the block texture energy exceeds the threshold for two out of three bands, then the block is considered to be highly textured. Similarly, if two out of three band energies fall below the thresholds $T_L(B)$, then the corresponding block is considered to be low in texture.

Each host image DCT block is thus classified into one of highly textured, normal, or low textured block, and the texture block factor γ is appropriately set. In the experiments below, the following parameter values are used:

$$T_H(B) = \frac{4}{3}, \forall B; T_L(B) = \frac{3}{4}; \gamma(\text{high}) = 2; \gamma(\text{normal}) = 0; \gamma(\text{low}) = -2.$$

5 DATA EMBEDDING

We can now summarize the various steps in the embedding procedure. Figure 5 gives the details of the encoder block.

1. The host and signature images are transformed to the DCT domain. A block size of 8×8 is used in the experiments below.
2. Each block of 8×8 host image pixels is analyzed for its texture content and the corresponding texture block factor γ is computed.



FIGURE 6. Test images

3. The signature coefficients are quantized according to the signature quantization matrix and the resulting quantized coefficients are encoded using lattice codes. The lattice codes are so chosen that the code vectors contain only ± 1 or zeros.
4. The signature codes are then appropriately scaled using the total scale factor $\delta = \alpha + \gamma$ and the JPEG quantization matrix¹¹. The JPEG quantization matrix helps in renormalizing the code vectors so that they have a similar dynamic range as a typical DCT block. Note that $\delta \geq 0$, which in turn constraints the choice of α and γ .
5. The selected host coefficients are then replaced by the scaled signature codes and combined with the original (unaltered) DCT coefficients to form a fused block of DCT coefficients. Note that more than one host coefficient is needed to encode a single signature code.
6. The fused coefficients are then inverse transformed to give an embedded image.

As discussed earlier, the choice of signature quantization matrix affects the quantity and quality of the embedded data. Choice of the scale parameter α depends on the application. A larger value for α results in a more robust embedding at the cost of quality of the embedded image, i.e., there could be perceivable distortions in the embedded image. A smaller α may result in poor quality recovered signature when there is a significant compression of the embedded image.

6 EXPERIMENTAL RESULTS AND DISCUSSIONS

Figure 6 shows the host and signature images used in the experiments. We use two different sizes for the host image: For embedding using the signature quantization matrix of Figure 3, a 256x256 host image is used, resulting in 25% data embedding. A 512x512 host image is used with the quantization matrix of Figure 4.



(a) Embedded without texture masking
 $(\alpha = 5, 89\%, \text{PSNR } 26.8\text{dB})$

(b) Embedded with texture masking
 $(\alpha = 5, \gamma = 2, 89\%, \text{PSNR } 29.4\text{dB})$

FIGURE 7. Watermarked images with and without texture masking. The signature quantization matrix shown in Figure 4 is used. Host image is 512x512 pixels and the signature image is 128x128 pixels. Notice the visible distortions in the sky region in (a).

Figure 7 shows the embedded images with and without texture masking. The signature quantization matrix shown in Figure 4 is used for this. From Figure 7(b), it is clearly seen that texture masking reduces visible distortions in regions that are *flat*, as in the sky part of the image.

Figure 8 shows recovered host and signature images for two different quantizations of the signature data, using texture masking. In this case, the embedded images are lossy compressed by JPEG to 89%. Obviously, the quantization matrix of Figure 4 yields better results than the one shown in Figure 3 at the cost of more host bits per signature coefficient.

Finally, Figure 9 shows the quality of the embedded and recovered images using the PSNR as a measure. It is clear from these graphs that one can achieve better quality embedding using the quantization matrix of Figure 4 at the cost of lower bit rate for the hidden data. Even at 25% embedding, one can recover visually acceptable quality results for upto 90% lossy compression using JPEG.

6.1 Discussions

We have proposed a robust data hiding technique for embedding images in images. A key component of the scheme is the use of multidimensional lattice codes for encoding signature image coefficients before inserting them into the host image DCT coefficients. Texture masking is used to reduce distortions in the embedded image by adaptively controlling the weights associated with the hidden data. The hidden signature data can be recovered in the absence of the original host image. Experimental results show that this method is robust to lossy image compression using JPEG. One can trade-off quantity for quality of the embedded image by choosing appropriate signature quantization matrices.



(b) Signature image recovered from Figure 7(b) (PSNR 31.7dB)

(a) Host image recovered from Figure 7(b) (PSNR 36.8dB)



(c) Embedded image (256x256) ($\alpha=5$, 89%, PSNR 22.7dB)



(d) Host image recovered from (c) (PSNR 31.4 dB)



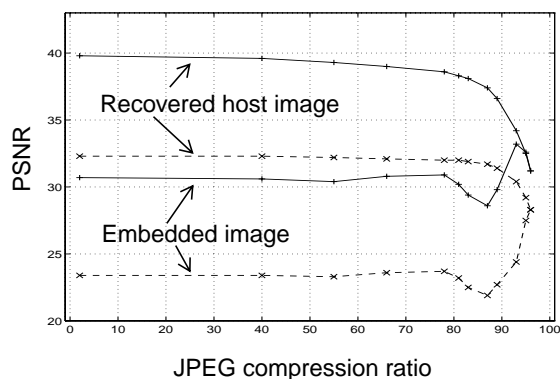
(e) Signature image recovered from (c) (PSNR 22.2dB)

FIGURE 8. Data embedding and recovery at two different bit rates. (a), (b) show results at 6% embedding; (c), (d) and (e) show the results for embedding signature data which is 25% of the host image.

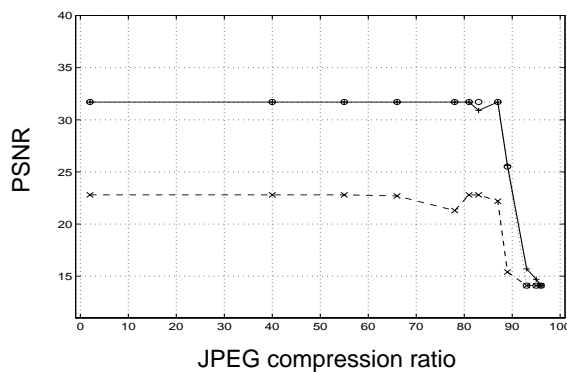
Acknowledgments: This work was supported in part by grants from NSF (award #94-1130 and #97-04785).

7 REFERENCES

- [1] I. J. Cox, J. Killian, T. Leighton, and T. Shamoan, "A secure Robust watermark for Multimedia," *IEEE Trans. Image Processing*, Vol. 6, no. 12, pp. 1673-1687, December 1997.
- [2] M. D. Swanson, M. Kobayashi and A. H. Tewfik, "Multimedia Data-Embedding and Watermarking Technologies," *Proceedings of the IEEE*, Vol. 86, no. 6, pp. 1064-1087, June 1998.
- [3] R. G. van Schyndel, A. Z. Tirkel and C. F. Osborne, "A Digital Watermark," *Proc. IEEE International Conference of Image Processing*, Vol. 2, pp. 86-90, Austin, Texas, November 1994.



(a) Watermarked and Recovered host images



(b) Recovered signature image

FIGURE 9. PSNR of embedded, recovered host and signature images (with scale factor 5) for different lossy JPEG compression factors. The solid lines are for 6% embedding using the quantization matrix of Figure 4. The dashed line shows the results at 25% embedding using the quantization matrix of Figure 3. Obviously, there is a clear trade-off between the quantity and quality of embedding.

- [4] J. J. Chae, D. Mukherjee and B. S. Manjunath, "A Robust Data Hiding Technique using Multidimensional Lattices," *Proceedings of the IEEE Forum on Research and Technology Advances in Digital Libraries*, pp. 319-326, Santa Barbara, April 1998.
- [5] A. Piva, M. Barni, F. Barolini and V. Cappellini, "DCT-based Watermark Recovering without Resorting to the Uncorrupted Original Image," *Proc. of IEEE International Conference of Image Processing*, Vol. 1, pp. 520-523, Santa Barbara, California, October 1997.
- [6] B. Tao and B. Dickenson, "Adaptive Watermarking in the DCT Domain," *Proc. of Intl. Conf. Accoustics, Speech and Signal Processing (ICASSP '97)*, Vol. 4, pp. 2985-2988, Munich, Germany, April 1997.
- [7] M. D. Swanson, B. Zhu and A. H. Tewfik, "Data Hiding for Video-in-Video," *Proc. IEEE International Conference of Image Conference*, Vol. 2, pp. 676-679, Santa Barbara, California, October 1997.
- [8] D. Mukherjee, J. J. Chae, and S. K. Mitra, "A Source and Channel Coding Approach to Data Hiding with Application to Hiding Speech in Video," *Proc. of IEEE International Conference of Image Processing*, Vol. 1, pp. 348-352, Chicago, Illinois, October 1998.
- [9] H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups*, Second edition, Springer-Verlag, New York, 1991.
- [10] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, Boston, 1992.
- [11] G. K. Wallage, "The JPEG Still Picture Compression Standard," *Communication of the ACM*, Vol. 34, no. 4, pp. 31-44, April, 1991.