Data Hiding in Video

J. J. Chae and B. S. Manjunath
Department of Electrical and Computer Engineering
University of California, Santa Barbara, CA 93106-9560
Email: chaejj, manj@iplab.ece.ucsb.edu

Abstract

We propose a video data embedding scheme in which the embedded signature data is reconstructed without knowing the original host video. The proposed method enables high rate of data embedding and is robust to motion compensated coding, such as MPEG-2. Embedding is based on texture masking and utilizes a multi-dimensional lattice structure for encoding signature information. Signature data is embedded in individual video frames using the block DCT. The embedded frames are then MPEG-2 coded. At the receiver, both the host and signature images are recovered from the embedded bit stream. We present examples of embedding image and video in video.

Keywords*:* data hiding, digital watermarking, multi-dimensional lattice structure.

1 Introduction

The internet and the world wide web have revolutionalized the way in which digital data is distributed. The widespread and easy access to multimedia content has motivated development of technologies for digital steganography or data hiding, with emphasis on access control, authentication, and copyright protection. Steganography deals with information hiding, as opposed to encryption. Much of the recent work in data hiding is about copyright protection of multimedia data. This is also referred to as digital watermarking.

Digital watermarking for copyright protection typically require very few bits, of the order of 1% or less of the host data size. These watermarks could be alpha-numeric characters, or could be multimedia data as well. One of the main objectives of this watermarking is to be able to identify the rightful owners by authenticating the watermarks. As such, it is desirable that the methods of embedding and extracting digital watermarks are resistant to typical signal processing operations, such as compression, and intentional attacks to remove the watermarks.

The focus of this paper differs from typical watermarking. We consider applications that require significantly larger amounts of data embedding. Examples of such applications include embedded control to track the use of a particular video clip in pay-per-view applications [1], hidden communications, smart images/video that can self-correct under intentional attacks, to mention a few. The capability to hide large amounts of data will also enable robust hiding of digital watermarks by introducing redundancies in the data. We use the term data hiding to distinguish such applications/ techniques from traditional watermarking. As such, the requirements for data hiding differ from those of watermarking. For example, while transparent or visible watermarks are acceptable in many cases, hidden data for control or secure communication need to be perceptually invisible.

The following terminology is used in this paper. The *signature* or *message data* is the data that we would like to embed or conceal. The *source data* is used to hide the signature data; we often refer to the source as the *host data*. After embedding a signature in to a host, we get the *watermarked* or *embedded data*. The *recovered data*, also referred to as the *reconstructed data*, is the signature that is extracted from the embedded data.

1.1 Previous Work

One of the early techniques for watermarking is the spread spectrum method proposed by Cox *et al.* [2]. The basic idea is to distribute the message or signature information over a wide range of frequencies of the host data. Many researchers have used the discrete cosine or the discrete wavelet transform coefficients to embed the signature data. While much of the initial work was on watermarking image data [3,4,5], recently several methods have been proposed for embedding audio and video information in video sequences. For example, Swanson *et al.* [6] proposed a data hiding algorithm to embed compressed video and audio data into video. The message data is embedded in the DCT domain, by modifying the projections of the 8x8 host block DCT coefficients. The data hiding rate is two bits per 8x8 block. The authors demonstrate robustness to additive Gaussian noise and motion JPEG compression. More recently, Mukherjee et al. [7] present a technique for hiding audio in video. They use multidimensional lattice structures to embed the a 8KHz speech signal, and the data hiding rate is about 1%.

In this paper, we describe a data hiding technique and demonstrate its robustness to MPEG coding of the embedded video. A schematic of our embedding scheme is shown in Figure 1. A key component of this scheme is the use of multidimensional lattices [9,10]. The signature image and host video frames are transformed using the 8x8 block DCT. The signature coefficients are quantized and 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 video frame blocks. The embedded video frames are then MPEG compressed, and the signature data is recovered from the lossy compressed video.

In the next section we describe the texture masking procedure. In texture masking, the strength of the signature signal is varied in proprtion to the local texture content of the host data. The signature image quantization is explained in Section 3. Section 4 details the steps in data embedding, and Section 5 describes the application to embedding in video and concludes with some experimental results.

2 Texture Masking

The human visual system is more sensitive to the changes in low frequency regions than in highly textured regions. Thus, insertions in the textured regions is less likely to result in visible distortions compared to less textured regions. Selective visual masking can be used to make the embeddings locally adaptive. For example, in [6] the authors use a model for frequency masking. This model predicts the detection threshold at a frequency *f* given the masking frequency f_m and local contrast c_m.

We suggest an alternative texture masking scheme that determines the amount of signature data to embed for each 8x8 host DC block. A scale factor γ controls the amount of inserted signature data. For textured regions this scale factor is kept low, where as for texture regions this is set to a higher value. Since the decisions are made in a 8x8 window, estimation of γ is quite robust and resistant to signal compression. The advantage is that at the decoding end the scale parameter can be directly computed from the received (embedded) signal. This is particularly important since we are assuming that the original host data is not available for reconstruction.

Consider a host 8x8 block and a one level wavelet decomposition of the block. Let *B={LH, HL, HH}* be the set of subbands. A Haar wavelet decomposition is used in our experiments. For a $b \in B$, Let $\mu_W(b)$ be the average energy in band b of the host image after a one level decomposition. Let $\mu_D(b)$ be the average energy in band b for the block under consideration. Define the block texture energy to be

$$\mu_T(b) = \frac{\mu_D(b)}{\mu_W(\mathrm{b})} \tag{1}$$

If $\mu_T(b)$ exceeds a given high 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 a low threshold $T_L(b)$, 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:

$$T_H(b) = \frac{4}{3}, \forall b \in B \tag{2}$$

$$T_L(b) = \frac{3}{4}, \forall b \in B \tag{3}$$

$$\gamma(\mathrm{high}) = 2\,,\ \gamma(\mathrm{normal}) = 0\,,\ \gamma(low) = -2\,. \tag{4}$$

3 Signature Image Quantization

There is clearly a trade-off between quantity of the data one can hide and quality of the embedded and reconstructed signals. 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/MPEG compression.

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 compared to the host data, thus controlling the quantity and quality of the embedded data. These quantized signature coefficients are then encoded using the multidimensional lattices and inserted into the host DCT coefficients.

Consider an 8 x 8 DCT coefficient matrix. The low frequency coefficients, obviously, require more bits than the high frequency ones. One such quantization matrix indicat-

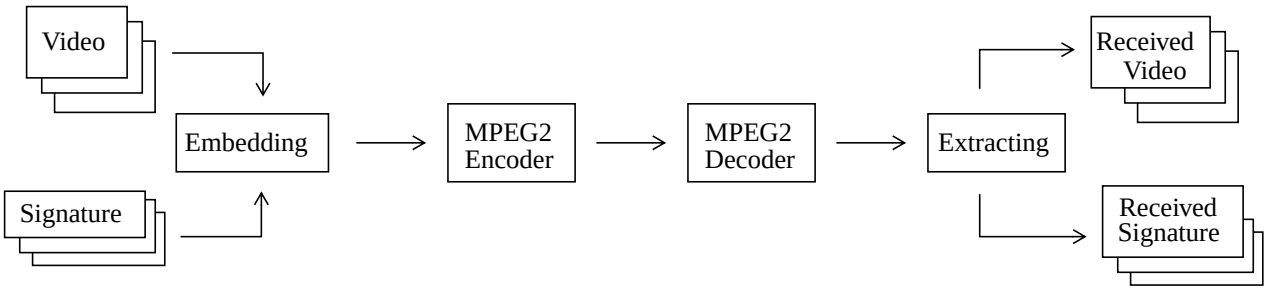
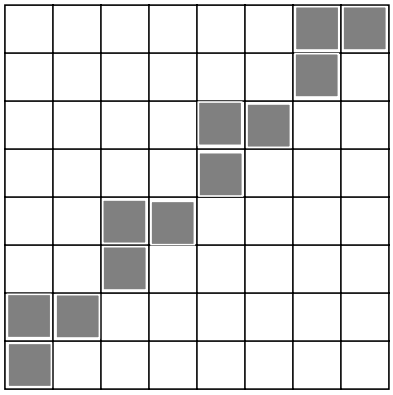


FIGURE 1. Schematic of video embedding technique

1232	1232	1232	342	342	342	48	48
1232	1232	342	342	342	48	48	0
1232	342	342	342	48	48	0	0
342	342	342	48	48	0	0	0
342	342	48	48	0	0	0	0
342	48	48	0	0	0	0	0
342	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(a) Signature quantization



(b) Selected coefficients for embedding

FIGURE 2. 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). A private key can be used to select the coefficients from the host image blocks.

ing the number of quantization levels for each of the 64 coefficients is shown in Figure 2(a). These quantized coefficients are embedded in a lattice structure [8], as described in [9,10]. For simplicity, we will consider only those shells in the lattice structure whose elements are $\{0, \pm 1, \pm 2\}$. 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 6 coefficients with this quantization (see Figure 2(a)), requiring 48 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 sixteen coefficients in the DCT matrix thus need 96 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 tweleve coefficients with this quantization, thus requiring 48 host coefficients.

The scheme outlined above thus needs a total of 192 host coefficients (6x8 + 66x6 + 12x4 = 192 coefficients) to embed the 64 DCT coefficients from one signature image block. The quantized coefficients are transformed to a lattice code, and the code is embedded into a partitioning of the host DCT block (shaded regions in Figure 2(b)).

4 Data Embedding

We now summarize the various steps in the embedding procedure. Figure 3 gives the details of the encoder block.

1. The host frame and signature image are transformed to the DCT domain. A block size of 8x8 is used in the experiments below.
2. Each block of 8x8 host frame pixels is analyzed for its texture content and the corresponding texture block factor γ is computed.
3. The signature coefficients are quantized according to the signature quantization matrix and the resulting quantized coefficients are encoded using lattice codes.
4. The signature codes are then appropriately scaled using the total scale factor $\delta = \alpha + \gamma$ and the JPEG quantization matrix. The JPEG quantization matrix helps renormalize the code vectors so that their dynamic range is similar to that of a typical DCT block. Note that $\delta \geq 0$.
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 than more than one host coefficient is needed to encode a single signature code. A private key can be used to select the ordering of the host/ signature blocks as well as in selecting the coefficients for embedding.
6. The fused coefficients are then inverse transformed to produce an embedded frame.

As discussed earlier, the choice of signature quantization matrix affects the quantity and quality of the embedded data. The choice of the scale parameter α depends on the application. A larger value for α results in an embedding which is more robust but might also result in loss of quality of the

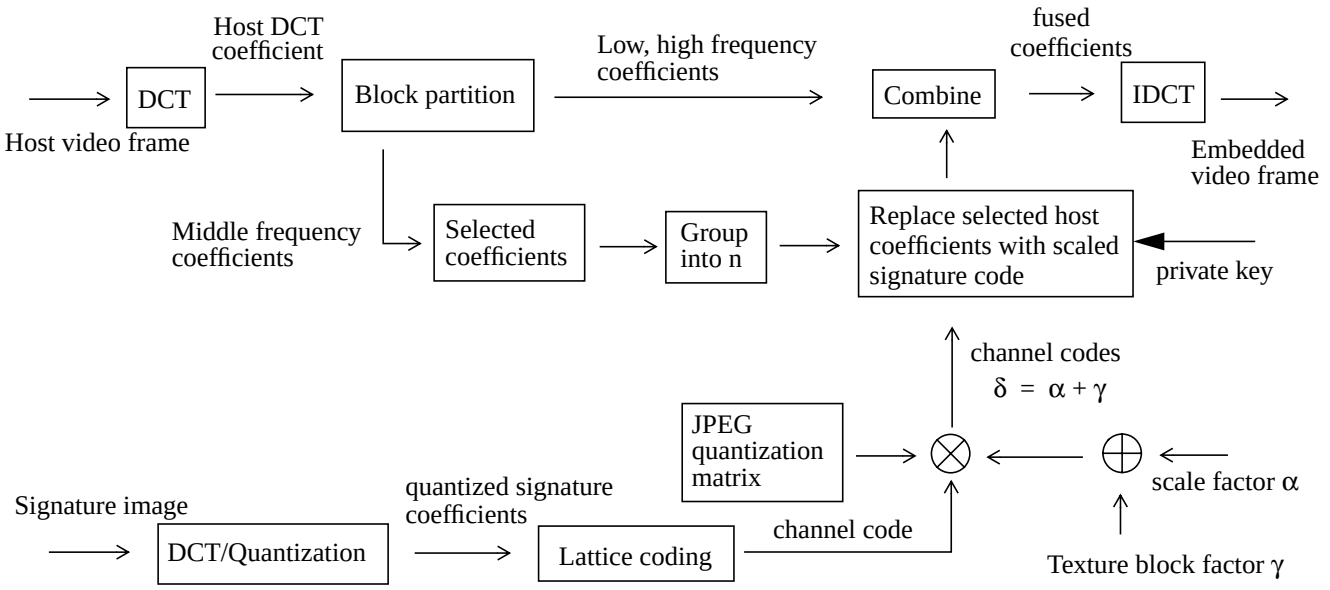


FIGURE 3. A schematic of the encoder in Figure 1

(a) Host frame 5 (QCIF, 176x144)

(b) Signature image (40x32: double size display)

Figure 4: test sources. (a) The Y component of car video frame 5. (b) The signature image (part of UCSB logo image).

embedded image, i.e. there could be perceivable distortions in the embedded video frame image. A smaller α may result in poor quality recovered signature especially when the embedded frame is subject to significant compression. The host frame and signature image are recovered following essentially an inverse sequence of operations.

5 Embedding in Video

Since a video can be viewed as a sequence of still images, video watermarking can be viewed simply as an extension of image watermarking. We use the Y component of a YUV color space representation for data hiding. This minimizes the color distortion in the embedded video.

Figure 4 shows samples of the test images. A host video frame is shown in Figure 4(a) and a signature image is shown in Figure 4(b). Note that 16 host video DCT blocks are required to embed one signature 8x8 DCT block.

To demonstrate the robustness to MPEG compression, we embed the signature image of Figure 4(b) into every frame of the host video sequence, and then compress the embedded video using MPEG2 at 600 Kbps, 30 frames per second. Figure 5(a) shows frame#5 of the sequence, reconstructed from the MPEG2 compressed video. Figure 5(b) shows the embedded frame#5 and Figure 5(c) shows this frame after MPEG2 coding/decoding. Figure 5(d) shows the reconstructed host frame from (c), and (e) shows the reconstructed signature from (c). Figure 5(f),(g) show the signature images retrieved from video frame#4 and #7 (P-frames in the MPEG2 coded video).

Direct embedding of video in video results in poor quality reconstructions of the embedded video. However, it is possible to modify the signature video prior to embedding such that the embedding and recovery are robust to MPEG compression. Figure 6 shows some preliminary results [11]. Figure 6(a) and (b) show the first frame of a host and signature video sequence, respectively. Figure 6(c) shows the watermarked frame and Figure 6(d) shows the reconstructed frame from the MPEG2 compressed video. The bit rate for MPEG2 was chosen to be 2 Mbps. The original, embedded, and reconstructed sequences are available on the web at http://vivaldi.ece.ucsb.edu.

In summary, we have presented a technique for hiding data in images and video. Compared to other methods, the proposed method can embed larger amounts of data and signature data can be recovered even under MPEG compression. Our current work is focussed on loss-less recovery of the signature when the embedded data undergoes lossy compression, and our preliminary results are quite encouraging. Loss-less recovery is important in embedding control or other binary data such as encrypted or encoded messages.

(a) Frame 5 after MPEG2 coding (PSNR 38.7dB)

(b) Frame 5 with scale factor 7 embedding (PSNR 30.8dB)

(c) Frame 5 after MPEG2 coding from embedding (b) (PSNR 27.8dB)

(d) retrieved Frame 5 after extracting from (c) (PSNR 35.5dB)

(e) Retrieved signature image from (c) (B type: PSNR 24.8dB)

(f) Retrieved signature image after MPEG coding from embedding frame 4 (P type: PSNR 35.1dB)

(g) Retrieved signature image after MPEG coding from embedding frame 7 (P type: PSNR 19.4dB)

Figure 5: (a) a B-frame from the car video sequence, MPEG-2 compressed at 600Kbps, 30 frames/second. (b) embedded frame, (c) embedded and after MPEG-2 compression, (d) reconstructed host frame, (e) recovered signature image from (c). (f), (g) two other examples, from embedded MPEG-2 P-frames.

(a) Host video frame # 0 (size: 352x240)

(b) Signature video frame # 0 (size: 352x240)

(c) Watermarked frame # 0 (PSNR 31.5dB)

(d) Recovered signature frame # 0 (PSNR: 45dB)

Figure 6: Video in Video embedding

Acknowledgments

This work was supported in part by a grant from NSF (award #97-04785). We would like to thank Dr. Mukherjee for many useful discussions and S. Newsam for his help in preparing this paper.

6 References

[1] 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.

[2] I. J. Cox, J. Killian, T. Leighton, and T. Shamoon, "A secure Robust watermark for Multimedia," *IEEE Trans. Image Processing*, Vol. 6. no. 12, pp. 1673-1687, December 1997.

[3] F. Hartung and B. Girod, "Watermarking of MPEG-2 encoded video without decoding and re-encoding," *Proceeding of SPIE EI '97, Multimedia Computing and Networking*, Vol. 3020, pp. 264-274, San Jose, California, January, 1999.

[4] L. Qiao and K. Nahrstedt, "Watermarking Methods for MPEG Encoded Video: Towards Resolving Rightful Ownership," *Proceedings of IEEE International Conference of Multimedia Computing and Systems*, pp. 276-285, Austin, June, 1998.

[5] B. Tao and B. Dickenson, "Adaptive Watermarking in the DCT Domain," *Proc. of Intl. Cond. Accoustics, Speech and Signal Processing (ICASSP '97)*, Vol. 4, pp. 2985-2988, Munich, Germany, April 1997.

[6] M. D. Swanson, B. Zhu and A. H. Tewfik, "Data Hiding for Video-in-Video," *Proceedings of IEEE International Conference of Image Processing (ICIP '97)*, Vol. 2, pp. 676-679, Santa Barbara, California, October, 1997.

[7] 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," *Proceeding of IEEE ICIP '98*, Vol. 1, pp. 348-352, Chicago, October, 1998.

[8] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups*, Second edition, Springer-Verlag, New York, 1991.

[9] J. J. Chae and B. S. Manjunath, "A Technique for Image Data Hiding and Reconstruction without Host Image," *to appear in the Proceeding of SPIE EI '99, Security and Watermarking of Multimedia Contents*, San Jose, California, January, 1999.

[10] J. J. Chae, D. Mukherjee and B. S. Manjunath, "Color Image Embedding using Multidimensional Lattice Structures," *Proceedings of IEEE International Conference of Image Processing (ICIP '98)*, Vol. 1, pp. 460-464, Chicago, Illinois, October, 1998.

[11] J. J. Chae, *Robust Techniques for Hiding Data in Images and Video*, Ph.D Dissertation, UCSB, June, 1999.