Abstract
Data hiding and watermarking in digital images and raw video have wide literature. This paper targets the internal dynamics of video compression, specifically the motion estimation stage. We have chosen this stage because its contents are processed internally during the video encoding/decoding which makes it hard to be detected by image steganalysis methods and is lossless coded, thus it is not prone to quantization distortions. In the literature, most work applied on data hiding in motion vectors relies on changing the motion vectors based on their attributes such as their magnitude, phase angle, etc. The data bits of the message are hidden in some of the motion vectors whose magnitude is above a predefined threshold, and are called candidate motion vectors (CMVs). A single bit is hidden in the least significant bit of the larger component of each CMV. The data is encoded as a region where the motion estimation is only allowed to generate motion vectors in that specified region. Using the variable macro block sizes, the authors have used every 2 bits from the message bit stream to select one of the four sizes for the motion estimation process. Here we embed the data in video using the phase angle between two consecutive CMV. These CMV are selected based on the magnitude of the motion vectors. The message bit stream is encoded as phase angle difference in sectors between CMV.
1 Introduction

Data hiding and watermarking in digital images and raw video have wide literature. This paper targets the internal dynamics of video compression, specifically the motion estimation stage. We have chosen this stage because its contents are processed internally during the video encoding/decoding which makes it hard to be detected by image steganalysis methods and is lossless coded, thus it is not prone to quantization distortions.

In the literature, most work applied on data hiding in motion vectors relies on changing the motion vectors based on their attributes such as their magnitude, phase angle, etc. The data bits of the message are hidden in some of the motion vectors whose magnitude is above a predefined threshold, and are called candidate motion vectors (CMVs).

A single bit is hidden in the least significant bit of the larger component of each CMV. In fig 4, the data is encoded as a region where the motion estimation is only allowed to generate motion vectors in that specified region. Using the variable macro block sizes, the authors in used every 2 bits from the message bit stream to select one of the four sizes for the motion estimation process. Here we embed the data in video using the phase angle between two consecutive CMV. These CMV are selected based on the magnitude of the motion vectors. The message bit stream is encoded as phase angle difference in sectors between CMV.

The block matching is constrained to search within the selected sector for a magnitude to be larger than the predefined threshold. The methods focused on finding a direct reversible way to identify the CMV at the decoder and thus relied on the attributes of the motion vectors. In this paper, we take a different approach directed towards achieving a minimum distortion to the prediction error and the data size overhead. This approach is based on the associated prediction error and we are faced by the difficulty of dealing with the nonlinear quantization process; thus we use an adaptive threshold as discussed in Section IV. We overview the terms of video compression and decompression. The problem definition is given along with the evaluation criteria used in this paper. Our proposed method is given and followed by the results and analysis.

2 Previous Method

2.1 High-Capacity Data Hiding in MPEG-2 Compressed Video

Digital watermarking, or information hiding, refers to techniques for embedding additional data in host media. Most of the previous research has focused on still image watermarking. Although video watermarking has more potential for commercial applications, less research has been conducted on high capacity data hiding in video streams. Ancillary data embedded in a video stream can carry information about the content itself, low-level descriptors for video indexing, retrieval and segmentation.

According to the bit to be embedded, a motion vector \( v \) is selected from nine neighbouring positions at half-pixel accuracy including \( v_{opt} \). That is, if the bit to be embedded is '0', \( v \) is extracted from the search area \( v_{opt} + \delta \cdot v_0 \), where \( \delta \cdot v_0 \in \{(0.5, 0.5), (0, 0), (0.5, -0.5), (-0.5, 0.5), (-0.5, -0.5)\} \) and (0, 0) represents the integer per position of \( v_{opt} \).

2.2 Embedding Process

The watermark is embedded in these AC components by applying the following modulation rule:
I. To embed the bit '1', the value of the selected AC component is changed to the nearest even number.
II. To embed bit '0', the value of the selected AC component is changed to the nearest odd number.

![Diagram](image)

**Figure 2.1 Embedding information in I frames of the MPEG-2 compressed video**

Since embedding the same amount of data in different I frames (or in different macro blocks of the same I frame) has different effect on the introduced distortion, adaptive masking is applied. Frames in the Group of Pictures (GOP) which the largest motion entropy are skipped. Pan B macro blocks within a I frame are also sensitive to distortions.

### 2.3 Data Hiding Procedure

Privacy data preservation demands high embedding capacity with minimal perceptual distortion. As privacy protection is typically applied to surveillance video which needs to be stored for a long period of time, the video must be compressed to minimize the amount of storage space needed. Our earlier results showed a disproportional increase in the compressed bit rate after the data hiding process.

### 2.4 Experiments

We tested our rate distortion based data hiding algorithms on two standard test sequences: “hall monitor” and “foreman”. Both sequences have 299 frames and are in CIF format (352×288). The “hall monitor” sequence is modified by in painting one of the persons with an adaptive estimate of the background. The image of the person being removed is compressed using a regular H.263 encoder with constant quantization.

This privacy information averages 2700 bits per frame but can fluctuate from 1000 bits to 20000 bits for individual frame. Results of the Exhaustive Search method and the Lagrangian approximation on the “hall monitor” sequence for QP=10 are shown in the first table.
2.5 Data Hiding In Video

One of the early techniques for watermarking is the spread spectrum method proposed by Cox et al. 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 transforms coefficients to embed the signature data while much of the initial work was on watermarking image data.

2.5.1 Data Embedding

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

I. The host frame and signature image are transformed to the DCT domain. A block size of 8x8 is used in the experiments below.

II. Each block of 8x8 host frame pixels is analyzed for its texture content and the corresponding texture block factor is computed.

III. The signature coefficients are quantized according to the signature quantization matrix and the resulting quantized coefficients are encoded using lattice codes.

IV. The signature codes are then appropriately scaled using the total scale factor and the JPEG quantization matrix. The JPEG quantization matrix helps renormalize the code vectors

2.5.2 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 above figure and a signature image is shown in below figure. Note that 16 host video DCT blocks are required to embed one signature 8x8 DCT block. Figure 2.3(d) shows the reconstructed host frame from (c), and (e) shows the reconstructed signature from (c). Figure 2.3 (f), (g) show the signature images retrieved from video frame#4 and #7 (P-frames in the MPEG2 coded video).

Figure2.2:234th frame of Hall monitor sequence after data hiding.
Figure 2.3 : Embedding in Video

3 Proposed Method

Data hiding and watermarking in digital images and raw video have wide literature. This project targets the internal dynamics of video compression, specifically the motion estimation stage. We have chosen this stage because its contents are processed internally during the video encoding/decoding which makes it hard to be detected by image steganalysis methods and is lossless coded, thus it is not prone to quantization distortions.

3.1 Block Diagram

Cover video

Video Sequence

Scene change Detector

Message Embedded

Video Steganalysis

Stego video

Key & Secret msg

Fig3.1 Block diagram
3.2 Proposed Method Algorithm

Our data-hiding algorithm is applied at the encoder side, uses the regular pair \((d,E')\) produced, tampers \(d\) to become \(d_h\), and thus replaces them by the pair \((d^h,E^h)\) for each P and B-frame in the GOP as shown in Algorithm 1. The secret message is organized as a bitstream \(m\), \(0 < k < K\) : message length. A subset of \(d\) is selected to be the CMV \(d\). The selection of \(d'(line 6 of Algorithm 1)\) is performed if their associated macro block \(B_{i,j}\) prediction error measured in PSNR is below an initial threshold value \(t_{\max}\).

- **Algorithm 1: Data Hiding in GOP**

  Input: message bitstream \(m\), GOP\((d,E)\), \(k\), \(\tau_{\max}\), \(\tau_{\min}\)
  
  Output: Data embedded in the Encoded GOP\((d^h, E^h)\)
  
  1. For each P and B frame in the GOP do
  2. Initialize \(\tau_{key} = \tau_{\max}\);
  3. Simulate the decoder: decompress \(\tilde{E}\) to obtain \(E_r\);
  4. Repeat
  5. Set \(d^h = d\);
  6. Obtain the candidate motion vectors \(\tilde{d}_{i,j}(x) = \{d_{i,j}(x) : 10\log_{10}(b^2/\sum B_{i,j}E_r(x)) \leq \tau_{key}\}\);
  7. while \((k \leq K) \& \; V(i,j) \in \tilde{d}_{i,j}(x)\) do
  8. replace the least significant bit \(\text{LSB} (\tilde{x}_{i,j}) = m(k), \text{LSB} (\tilde{y}_{i,j}) = m(k+1)\);
  9. \(k = k+2\);
  10. if B frame then
  11. \(\text{LSB} (\tilde{x}_{i,j}) = m(k), \text{LSB} (\tilde{y}_{i,j}) = m(k+1)\);
  12. \(k = k+2\);
  13. end
  14. \(d^h_{i,j} = \tilde{d}_{i,j}\);
  15. end
  16. compute associated \(E^h(x)\) by suitable compensation using \((x + d^h(x))\);
  17. \([\text{key found}, \tau_{key}] \leftarrow \text{validate} (E^h, \tau_{key}, \tilde{d})\);
  18. Until key found or \(\tau_{key} = \tau_{\min}\);
  19. if not key found then
  20. \(\tau_{key} = -1\)
  21. end
  22. Hide \(\tau_{key}\) in I-frame or send on a separate channel;
  23. end

- **Algorithm 2: Validate \(\tau\)**

  Input: \(E^h\), \(\tau_{key}\), \(\tilde{d}\)
  
  Output: key found, \(\tau_{key}\)
  
  1. Compress \(E^h\) using JPEG compression to produce \(\tilde{E}^h\);
2 Decompress $\tilde{E}^h$ to obtain lossy $H_e^h$ ;
3 Set key found = True ;
4 While key found & (i,j) $\in \tilde{d}_{i,j}(x)$ do
5 if $10 \log_{10}(b/\sum B_{i,j}E^h_{e}(x)) > \tau_{key}$ then
6 key found = false;
7 decrement $\tau_{key}$;
8 end
9 end

The decoder receives the pair $(d^h, \tilde{E}^h)$ and it can decode $dh$ without loss and decompresses $E^h$ to obtain a lossy reconstructed version $E_{r}^h$. During normal operation for viewing the video, the decoder is able to reconstruct $prh$ or $brh$ by suitable compensation from reference frame(s) using $(x+d^h(x))$ and adding $E_{r}^h$ to it.

- **Algorithm 3: Data Extraction**

Input: GOP $(d^h, \tilde{E}^h), k$

Output: message bitstream $m$

1 Extract the threshold $\tau_{key}$ for all frames in GOP from I-frame or use them from other channel;
2 For each P & B frames in the GOP do
3 Decompress $\tilde{E}^h$ to obtain $E_r^h$, and identify the candidate motion vectors: $\tilde{d}_{i,j}(x) = \{d^h_{i,j}(x) : 10 \log_{10}(b/\sum B_{i,j}E^h_{e}(x)) \leq \tau_{key}\}$
4 For each (i,j)$\in \tilde{d}_{i,j}(x)$ do
5 Extract 2 message bits $m(k) = \text{LSB}($ $\tilde{x}_{i,j})$, $m(k+1) = \text{LSB}($ $\tilde{y}_{i,j})$;
6 $k=k+2$;
7 if B-frame then
8 Extract from backward compensation motion vectors 2 message bits $m(k) = \text{LSB}($ $\tilde{x}_{i,j})$, $m(k+1) = \text{LSB}($ $\tilde{y}_{i,j})$;
9 $k=k+2$;
10 end
11 end
12 End

**4 Results And Discussion**

We implemented the hiding and extraction Algorithms 1, 2, and 3 and integrated them to the MPEG-2 encoder and decoder operation. The parameters of our experiments, presented in this section, are: macro block size $b=16$, motion vector representation bits $n=5$. We used both the fast three-steps and exhaustive search motion estimation algorithms with half pixel accuracy. Each test video sequence is organized into consecutive GOP organized as [I,B,P,B,B,P,B,B]. Our algorithm may hide a maximum of $2 * B / 8$ bytes per P-frame and $4 * B / 8$ per B-frame. Analyzing the PSNR values of the prediction error $E_r$ for all
sequences, we set $t_{max} = 60$, $t_{min} = 20$ dB for P-frames, and $t_{max} = 40$, $t_{min} = 15$ dB for B-frames. We evaluated our algorithm and compared it to an attribute-based method which is dependent on a threshold $T$ of the magnitude of the motion vectors.

Figure 4.1: Embedding Process

Figure 4.2: Results for Extracting Process
5 Conclusion

We proposed a new data-hiding method in the motion vectors of MPEG-2 compressed video. Unlike most data-hiding methods in the motion vectors that rely their selection on attributes of the motion vectors, we chose a different approach that selects those motion vectors whose associated macro blocks prediction error is high (low PSNR) to be the candidates for hiding a bit in each of their horizontal and vertical components.

References


Authors Profile

JAYAMPU MANOHAR, M.Tech., Working as an Associate Professor in the Department of ECE at Jagan’s College of Engineering & Technology, Chawatapalem, Nellore ,A.P, India. He is having 7 years of Experience in Teaching. His area of research is Low Power VLSI Design, Image Processing and Communication Systems.

Dr.K.V.Ramanaiah, M.Tech, Ph.D Working as Professor in the Department of ECE at Yogi Veman University, Proddutur, Kadapa. A.P, India. He is having 20 years of Experience in Teaching. His area of research is VLSI, Image Processing and Neural Networks.