Steganography in Compressed Video Stream

Changyong Xu	Xijian Ping	Tao Zhang
Department of Information Science, Zhengzhou Information Science and Technology Institute, 450002, P.R.China	*Department of Information Science, Zhengzhou Information Science and Technology Institute, 450002,P.R.China*	*National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, 100080, P.R.China*
chyong80@163.com	*pingxijian@yahoo.com.cn*	*brunda@sina.com*

Abstract

In this paper, a steganographic algorithm in MPEG compressed video stream was proposed. In each GOP, the control information for to facilitate data extraction was embedded in I frame, in P frames and B frames, the actually transmitted data were repeatedly embedded in motion vectors of macro-blocks that have larger moving speed, for to resist video processing. Data extraction was also performed in compressed video stream without requiring original video. On a GOP by GOP basis, control information in I frame should be extracted firstly, then the embedded data in P and B frames can be extracted based on the control information. Experimental results show that the proposed algorithm has the characteristics of little degrading the visual effect, larger embedding capacity and resisting video processing such as frame adding or frame dropping.

1. Introduction

Information hiding is a recently rapidly developed technique in the field of information security and has received significant attention from both industry and academia. It contains two main branches: digital watermarking and steganography. The former is mainly used for copyright protection of electronic products. while steganography, as a new way of covert communication, the main purpose is to convey data secretly by concealing the very existence of communication[1].

The carrier for steganography can be image, text, audio and video. Image is the most familiar carrier, but the limited size of image will inevitably restrict the capacity of embedding. In the case of requiring transmitting large number of secret messages, steganography in image will not satisfy the demand. So steganographic method that has higher embedding capacity need to be studied. Because digital video is composed of series of frames and has greater signal space, steganography in video will get large capacity. Furthermore, with the development of multimedia and stream media on the Internet, transmitting video on the Internet will not incur suspicion. Besides, the degradation of video quality cannot be observed only by naked eyes, for it may be aroused by video compression of lower quality. These reasons make it possible for us to securely hide data in video.

Steganography in video can be divided into two main classes. One is embedding data in uncompressed raw video, which is compressed later[2,3]. The other, which is more difficult, tries to embed data directly in compressed video stream[4,5,6]. The problem of the former is how to make the embedded message resist video compression. But because the video basically exists in the format of compression, the research of the latter is more significative. A steganographic algorithm for compressed video is introduced in this paper,, operating directly in compressed bit stream. In a GOP, control information is embedded in I frame, and in P frames and B frames, the data are repeatedly embedded in motion vectors of macro-blocks, for the purpose of resisting video processing.

2. Steganographic algorithm

2.1. Data embedding in P frame and B frame

The data stream of MPEG is mainly composed of head information, DCT encoded data stream and motion vector data stream. DCT encoded data stream is produced by intra-encoding to I frames, encoding P frames and B frames using motion compensation prediction technique produced motion vector data stream. In P frames each macro-block has one motion

Proceedings of the First International Conference on Innovative Computing, Information and Control (ICICIC'06)
0-7695-2616-0/06 $20.00 © 2006 **IEEE**

vector, while in B frames each has two motion vectors. All motion vectors are times of half-pixel. Steganographic algorithm should be designed based on these characteristics. In P frames and B frames, each macro-block has motion vector, so the data can be embedded in motion vectors. Furthermore, in MPEG video sequences, besides intra-frames, most frames are encoded using motion compensation prediction. So data embedding in motion vectors can adequately utilize the information of compressed video stream, and achieve higher embedding capacity.

Except those at scene changes or with fast motion, sequential frames in a video look similar. Due to this, it is possible to add or drop some frames, or change the order of adjacent frames, but not introducing much noticeable distortion. Since such manipulations can arise from common video processing, these problems must be taken into account when we design steganographic algorithm. By adding redundancy, that's to say, embedding data in a video clip repeatedly can solve these problems. This method can be named redundant embedding, and in our algorithm the method is adopted.

In the proposed algorithm, the data were not embedded in each motion vector of P frames and B frames, but the motion vectors with larger magnitude. The larger magnitude indicates the faster moving speed of the macro-blocks. In this case, the distortion introduced by data embedding is minimal comparing to modify all motion vectors include those with slight movement or even still. The details of data embedding are as follows. The method of modifying motion vectors is derived from the algorithm of Jun Zhang et al[6].

1) For a P frame or B frame, get the motion vector $PMV[i], 0<i<N_{MB}$ from the compressed video stream, where N_{MB} is the number of macro-blocks in this frame.

2) Calculate the magnitude of motion vector $|PMV[i]|=\sqrt{H^2[i]+V^2[i]}$, Where $H[i]$ is the horizontal component of motion vector in the *ith* macro-block, $V[i]$ is the vertical component.

3) Given threshold of the magnitude of motion vector is ε , we can select the embeddable macro-block $MB[i]=\begin{cases}1 & |PMV[i]|>\varepsilon \\ 0 & |PMV[i]|\leq\varepsilon\end{cases}$, where $MB[i]=1$ denotes the macro-block satisfies the condition, and can be used to embed data, $MB[i]=0$ denotes data were not embedded in this macro-block.

4) For all the P frames and B frames in a GOP, we do the same calculation, and achieve the number of eligible motion vectors. For each eligible motion vector, one bit can be embedded. Thus we can obtain the embedding capacity of a GOP, which will be embedded in I frame to facilitate data extraction. In order to resist video processing, redundant embedding is introduced. For computational simplicity, it is fulfilled by dividing the embeddable capacity of a GOP into four segments, and embedding same data to a GOP four times. The position of the start and the end of each segment should be marked down and embedded into I frame as a part of control information.

5) Embedding data in the selected macro-blocks. Firstly, we calculate the phase angle of motion vector $\theta[j]=\arctan\left(\frac{V[j]}{H[j]}\right)$, then based on the value of θ to determine the embedding scheme.

a. If θ is acute angle, less distortion will be introduced by modifying horizontal component of motion vector, so the data were embedded into horizontal component.

If $\mathrm{mod}(2*H[j],2)=mes(k)$, then $H_m[j]=H[j]$

If $\mathrm{mod}(2*H[j],2)\neq mes(k)$, then $H_m[j]=H[j]+0.5$

Where $H_m[j]$ is horizontal component of motion vector in the *jth* macro-block after embedding, the magnitude of modification to motion vector is 0.5, which is minimal. mes is the data to be embedded, and is also bit stream.

b. If θ is obtuse angle, less distortion will be introduced by modifying vertical component of motion vector, so the data will be embedded into vertical component.

If $\mathrm{mod}(2*V[j],2)=mes(k)$, then $V_m[j]=V[j]$

If $\mathrm{mod}(2*V[j],2)\neq mes(k)$, then $V_m[j]=V[j]+0.5$

Where $V_m[j]$ is vertical component of motion vector in the *jth* macro-block after embedding.

6) In a GOP, based on the embedding capacity we have calculated above and the position of the start and the end of each segment, the data were embedded repeatedly into each segment until reaching the end of GOP. At last, each GOP in the video sequence was embedded with data.

2.2. Embedding control information in I frame

I frame is the first frame in a GOP, data hiding in I frame is to facilitate the extraction of embedded data. The data that need to be actually transmitted were embedded in P frames and B frames, and the embedding is fulfilled by a slight modification to motion vectors. Furthermore, redundant embedding is adopted to increase the algorithm's security. So control information includes the embedding capacity of the GOP, the position of the start and the end of each segment after dividing the GOP.

Comparing to the data that embedded in P frames and B frames, the amount of control information is relatively small, but is critical and should be extracted accurately. Consequently, higher robustness is required. I frame is compressed using intra-frame DCT encoding, the algorithm is similar to JPEG, so data embedding in I frame is similar to the algorithm for still images. We embed the information in middle frequency of quantized DCT coefficients, for the sake of obtaining the tradeoff between imperceptibility and robustness. Firstly we should get the quantized DCT coefficients, which can be achieved by variable length decoding after the minimal parsing of the video stream. Then we embed control information into quantized DCT coefficients. The embedding is realized by modifying the least significant bits of quantized DCT coefficients[7]. The modified coefficients are variable length encoded and placed back into bit stream to form embedded bit stream of I frame.

3. Data extraction algorithm

Data extraction was accomplished on a GOP by GOP basis. For a GOP, the control information in I frame should be accurately extracted firstly. Variable length decoding is performed to obtain the quantized DCT coefficients and the middle frequency coefficients are selected. By taking the least significant bits from these coefficients, we can obtain control information.

Then the embedded data in P frames and B frames can be extracted based on the control information we have just obtained. The details are as follows.

1) Get the motion vector $PMV[i], 0<i<N_{MB}$ from the compressed video stream, where N_{MB} is the number of macro-blocks in this frame.

2) Calculate the magnitude of motion vector $|PMV[i]| = \sqrt{H^2[i]+V^2[i]}$, Where $H[i]$ and $V[i]$ are horizontal component and vertical component of motion vector in the *ith* macro-block respectively.

3) Based on the threshold of the magnitude of motion vector, we can select the embeddable macro-block $MB[i]=\begin{cases}1 & |PMV[i]|>\varepsilon \\ 0 & |PMV[i]|\le\varepsilon\end{cases}$, where $MB[i]=1$ and $MB[i]=0$ respectively denotes the macro-block has been embedded and not embedded data.

4) For the motion vectors that have been embedded data, calculate the phase angle $\theta[j]=\arctan\left(\frac{V_m[j]}{H_m[j]}\right)$.

a. If θ is acute angle, data was embedded into horizontal component, then $mes(k)=\mathrm{mod}(H_m[j]*2,2)$.

b. If θ is obtuse angle, data was embedded into vertical component, then $mes(k)=\mathrm{mod}(V_m[j]*2,2)$.

5) For all P frames and B frames, the calculation above was done until all embedded data in a GOP were extracted. Due to the adoption of redundant embedding, the data extracted from P frames and B frames were not exactly the data we want to convey, we should determine the data by control information that has extracted from I frame. Via weighted majority voting, i.e. larger weights assigned to the frames experiencing less distortion, the data we exactly want to convey by this GOP can be obtained. The amount of embedded data can also be examined. All these will increase the accuracy of data extraction.

6) For all GOPs, the calculation above was done. Link data extracted from P frames and P frames together, we can obtain the actually transmitted data.

4. Experimental results

In the experiment, a video sequence is selected as the test video. A text document was embedded into the video. One I frame, B frame and P frame in a GOP of this video before and after embedding and their differences are shown in Figure 1. The frames have been transformed to gray-scale format and the difference images have been gray-level stretched by 30 times for the purpose of display.

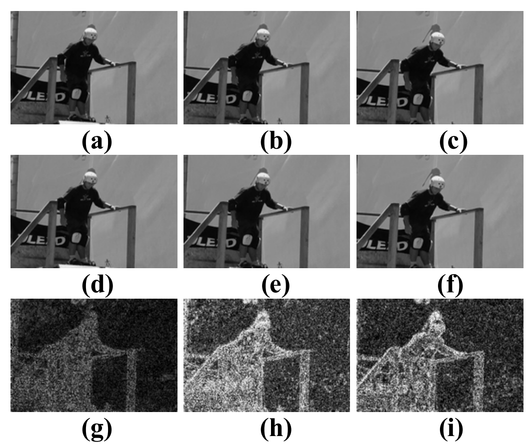


Figure 1. **(a) Original I frame. (b) Original B frame. (c) Original P frame. (d) I frame after embedding. (e) B frame after embedding. (f) P frame after embedding. (g) Stretched difference between (a) and (d). (h) Stretched difference between (b) and (e). (i) Stretched difference between (c) and (f).**

We used PSNR to quantitatively measure the degradation introduced by data embedding, it is listed in Table 1.

Table 1. **PSNR**

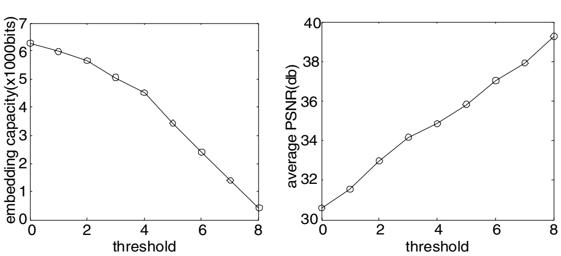
Frame	I frame	P frame	B frame
PSNR(db)	35.2273	34.6136	33.3175

In the following, we analyze the embedding capacity of the algorithm. The test video has 240 frames, in each GOP there are 12 frames, include one I frame, three P frames and eight B frames. The frame size is 352×240, given the size of macro-block is 16×16, thus in each frame there are 330 macro-blocks. Based on our algorithm, the embedding capacity of each GOP can be calculated. In Table 2, for a specified threshold of magnitude, the embedding capacity of a GOP was listed. It is should be noted because there are more motion vectors in B frames than in P frames and the number of B frames is more than P frames, the embedded bits in B frames is much more than that in P frames. The table only listed the data of one GOP. For different GOP, it may vary greatly. Due to redundant embedding, the number of actually embedded data in the GOP should be divided by four.

Table 2. **The amount of embedded data in a GOP**

Frame type	P	B
Number of macro-blocks	990	2640
Embedded data(bits)	537	4519

In the experiment, when we select different thresholds for the magnitude of motion vector, the result will be different. The smaller the threshold, the more embedding capacity we can get, and the larger the distortion introduced by data embedding. This is reasonable because the less modification to the video, the less distortion will be introduced. The relationship of them was shown in Figure 2. Therefore, in the practical application, we should select appropriate threshold based on the requirement, for the purpose of obtaining considerable embedding capacity, and at the same, keeping good perceptive effect of video quality as well.



(a) **(b)**

Figure 2. **(a) Relationship between embedding capacity of a GOP and threshold. (b) Relationship between average PSNR of all frames in a GOP and threshold.**

Moreover, the capability of resisting video processing of the proposed algorithm was tested. The experimental results show, since the adoption of redundant embedding, the data can still be extracted after frame adding and frame dropping been done to the video. But for complicated video processing technique, such as transcoding and format conversion, the embedded data cannot survive perfectly.

5. Conclusions

A steganographic algorithm for data embedding in video was proposed in this paper, operating directly in compressed bit stream. By embedding control information in I frame and redundant embedding in P and B frames, the capability of resisting video processing was achieved and a good balance between embedding capacity and security was obtained, but part of embedding capacity was sacrificed. So in the future, more effective methods should be taken into account to further increase the embedding capacity and enhance the security of the algorithm as well. Furthermore, modification was only done to motion vectors that have larger magnitude, it is equal to embedding data essentially in text areas and on edge, for smooth regions the modification is small or even zero, so high imperceptibility was achieved.

6. References

[1] F.A.P.Petitcolas, R.J.Anderson, M.G.Kuhn, "Information Hiding-A Survey", *Proceeding of the IEEE,* vol. 87, no. 7, June 1999, pp.1062-1078.

[2] J. J. Chae, B. S. Manjunath, "Data Hiding in Video", *Proceedings of the 6th IEEE International Conference on Image Processing,* 1999, pp.311-315.

[3] Melih Pazarci, Vadi Dipcin, "Data Embedding in Scrambled Digital Video", *Proceedings of the 8th IEEE International Symposium on Computers and Communication,* 2003, pp. 498-503.

[4] A. Giannoula, D. Hatzinakos, "Compressive Data Hiding for Video Signals", *Proceedings of International Conference on Image Processing,* 2003, pp. I529- I532.

[5] Giuseppe Caccia, Rosa Lancini, "Data Hiding in MPEG2 Bit Stream Domain", *Proceedings of International Conference on Trends in Communications,* 2001, pp.363-364.

[6] Jun Zhang, Jiegu Li, Ling Zhang, "Video Watermark Technique in Motion Vector", *Proceedings of XIV Brazilian Symposium on Computer Graphics and Image Processing,* 2001, pp.179-182.

[7] Westfeld A, "F5-A Steganographic Algorithm", *Proceeding of 4th International Conference on Information Hiding, LNCS 2137,* Springer-Verlag, 2001, pp.289-302.