ISSN:0975-9646

Effective Lossless Authentication in Video for Data Embedding Application

Prof .Deepali S.Chavan

Assistant Professor, Dept. of Computer Engineering and Information Technology,
Veermata Jijabai Technological Institute (VJTI) – Mumbai, India

***Abstract* — A video data embedding scheme in which the embedded signature data is reconstructed without knowing the original host video. Signature data is embedded in individual video frames using the block BinDCT. At the receiver, both the host and signature images are recovered from the embedded bit stream. Lossless authentication achieves the same goal with the advantage that the distortion can be erased if media authenticity is positively verified. Here, we propose a data hiding and extraction procedure for high resolution AVI (Audio Video Interleave) videos. Although AVI videos are large in size but it can be transmitted from source to target over network after processing the source video by using these Data Hiding and Extraction procedure securely. There are two different procedures, which are used here at the sender's end and receiver's end respectively. The procedures are used here as the key of Data Hiding and Extraction.**

Keywords- Authentication,BinDCT,Data Hiding.

I. INTRODUCTION

The internet and the World Wide Web have revolutionaries 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. 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.

Data embedding applications include embedded control to track the use of a particular video clip in pay-per-view applications [1]. A short media digest, such as the cryptographic hash, is embedded in the media itself rather than attached to it in a header or a separate file [2]. 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 wavelets transform coefficients to embed the signature data. While much of the initial work was on watermarking image data [3, 4]. A data hiding algorithm to embed compressed video and audio data into video [5].

II. PREVIOUS WORKS

As video file consist of several image sequence, so considering the data hiding technique of image will also apply for video data hiding we get:

A. *Least-significant bit modifications*

The most widely used technique to hide data, is the usage of the LSB. Although there are several disadvantages to this approach, the relative easiness to implement it, makes it a popular method. To hide a secret message inside an image, a proper cover image is needed. Because this method uses bits of each pixel in the image, it is necessary to use a lossless compression format, otherwise the hidden information will get lost in the transformations of a lossy compression algorithm. When using a 24 bit color image, a bit of each of the red, green and blue color Components can be used, so a total of 3 bits can be stored in each pixel. Thus, an 800 × 600 pixel image can contain a total amount of 1.440.000 bits (180.000 bytes) of secret data. For example, the following grid can be considered as 3 pixels of a 24 bit color image, using 9 bytes of memory:

(00100111 11101001 11001000)
(00100111 11001000 11101001)
(11001000 00100111 11101001)

When the character A, which binary value equals 10000001, is inserted, the Following grid results:

(00100111 11101000 11001000)
(00100110 11001000 11101000)
(11001000 00100111 11101001)

In this case, only three bits needed to be changed to insert the character successfully. On average, only half of the bits in an image will need to be modified to hide a secret message using the maximal cover size. The resulting changes that are made to the least significant bits are too small 1 to be recognized by the human eye, so the message is effectively hidden. While using a 24 bit image gives a relatively large amount of space to hide messages, it is also possible to use an 8 bit image as a cover source. Because of the smaller space and different properties, 8 bit images require a more careful approach. Where 24 bit images use three bytes to represent a pixel, an 8 bit image uses only one. Changing the LSB of that byte will result in a visible change of color, as another color in the available palette will be displayed. Therefore, the cover image needs to be selected more carefully and preferably be in greyscale, as the human eye will not detect the difference between different grey values as easy as with different colours.

B. *Masking and filtering*

Masking and filtering techniques, usually restricted to 24 bits or greyscale images, take a different approach to hiding a message. These methods are effectively similar to paper watermarks, creating markings in an image. This can be achieved for example by modifying the luminance of parts of the image. While masking does change the visible Properties of an image, it can be done in such a way that the human eye will not notice the anomalies. Since masking uses visible aspects of the image, it is more robust than LSB modification with respect to compression, cropping and different kinds of image processing. The information is not hidden at the ”noise” level but is inside the visible part of the image, which makes it more suitable than LSB modifications in case a lossy compression algorithm like JPEG is being used [2].

C. *Tranformatios*

A more complex way of hiding a secret inside an image comes with the use and modifications of discrete cosine transformations. Discrete cosine transformations (DST)), are used by the JPEG compression algorithm to transform successive 8 x 8 pixel blocks of the image. A simple pseudo-code algorithm to hide a message inside a JPEG image could look like this :

```
1. Input: Message, cover image
2. Output: Data hiding image
   containing message
   a. While data left to embed do
   b. Get next DCT coefficient
      from cover image
   c. If DCT not equal to 0 and
      DCT not equal to 1 then
        i. Get next LSB from
           message
       ii. Replace DCT LSB with
           message bit
   d. End if
   e. Insert DCT into Data hiding
      image
3. End while
```

III PROPOSED SYSTEM DESIGN AND IMPLEMENTATION

The main high resolution AVI fie is nothing but a sequence of high resolution image called frames. Initially we will like to stream the video and collect all the frames and system flow of proposed system as shown in (Figure 1). And also collect the following information:

- Starting frame: It indicates the frame from which the algorithm starts message embedding.
- Starting macro block: It indicates the macro block within the chosen frame from which the algorithm starts message embedding.
- Number of macro blocks: It indicates how many macro blocks within a frame are going to be used for data hiding. These macro blocks may be consecutive frame according to a predefined pattern. Apparently, the more the macro blocks we use, the higher the embedding capacity we get. Moreover, if the size of the message is fixed, this number will be fixed, too. Otherwise it can be dynamically changed.
- Frame period: It indicates the number of the interframes, which must pass, before the algorithm repeats the embedding. However, if the frame period is too small and the algorithm repeats the message very often, that might have an impact onto the coding efficiency of the encoder [4].

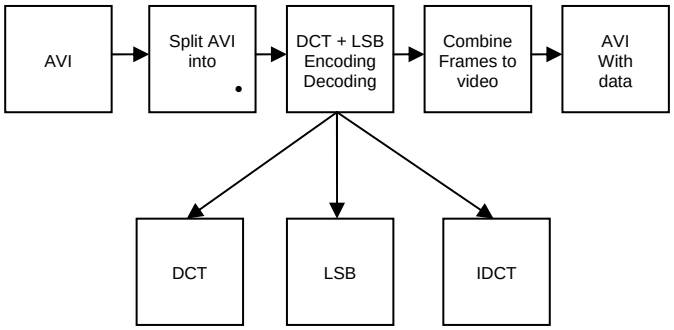

Figure 1: System flow for proposed system

IV BINDCT ALGORITHM

Linear transforms BinDCT algorithm is used in the first stage of the image compression system. The transform function can work with both lossless and lossy systems. The proposed binDCT is also based on the well. Known Chen.Wang plane rotation. based factorization of the DCT matrix. Properties of BinDCT:

1) Both the forward and the inverse transforms can be implemented using only binary shift and addition operations.

2) The idea of the scaled DCT is employed to reduce the complexity of the binDCT.

3) The binDCT inherits all desirable DCT characteristics such as high coding gain, no DC leakage, symmetric basis functions, and recursive construction.

4) The binDCT also inherits all lifting properties such as fast implementations, invertible integer-to-integer mapping, in-place computation, and low dynamic range.

General Structure of the binDCT

From the above results, we can obtain the general structure for the lifting. Based 4 point DCT, denoted by binDCT. The forward and inverse transforms are given in Figure3. The rotations of $\pi/4$ and $3\pi/8$ are both represented by 2 lifting steps. Note that the permuted scaled lifting structure is used for $3\pi/8$, and because of the butterflies, a scaling factor of 2 is introduced after the inverse transform, which can be compensated by a right Shift.

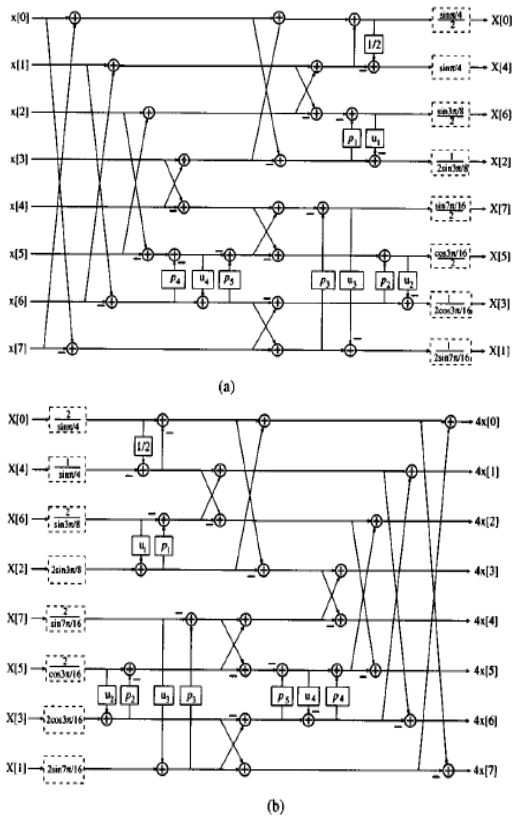

Figure 2: a) Forward Transform (b) Inverse transform

Table 1 tabulates the floating-point values of the parameters p and u in Figure3, as well as some of their dyadic approximations, which allow for fast implementations with only shift and addition operations.Table I tabulates the forward and inverse transform matrices of the binDCT-C7, without including the final scaling factors. With the help of Table I we calculate forward and inverse BinDCT.

binDCT-C7 Forward Transform Matrix							
1	1	1	1	1	1	1	1
15/16	101/128	35/64	1/4	-1/4	-35/64	-101/128	-15/16
3/4	1/2	-1/2	-3/4	-3/4	-1/2	1/2	3/4
1/2	3/32	-11/16	-1/2	1/2	11/16	-3/32	-1/2
1/2	-1/2	-1/2	1/2	1/2	-1/2	-1/2	1/2
1	-23/16	-1/8	1	-1	1/8	23/16	-1
1/2	-1	1	-1/2	-1/2	1	-1	1/2
1/4	-21/32	13/16	-1	1	-13/16	21/32	-1/4

(a)

binDCT-C7 Inverse Transform Matrix							
1/2	1	1	1	1	1/2	1/2	1/4
1/2	13/16	1/2	1/8	-1	-11/16	-3/4	-35/64
1/2	21/32	-1/2	-23/16	-1	-3/32	3/4	101/128
1/2	1/4	-1	-1	1	1/2	-1/2	-15/16
1/2	-1/4	-1	1	1	-1/2	-1/2	15/16
1/2	-21/32	-1/2	23/16	-1	3/32	3/4	-101/128
1/2	-13/16	1/2	-1/8	-1	11/16	-3/4	35/64
1/2	-1	1	-1	1	-1/2	1/2	-1/4

(b)

Table 1: bindct-c7 coefficients (a) Forward transform (b)Inverse transform

IV DATA EMBEDDING AND EXTRACTION PROCESS

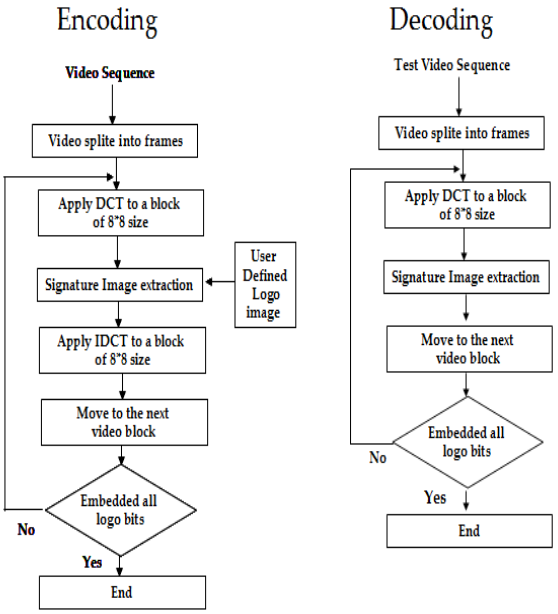

Figure 3: Encoding and Decoding flow chart

V 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. 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.

(a) (b)

Figure 4 (a) parrot. AVI Frame # 4 (b) Signature text

A host video frame is shown in Figure 6 and a signature data is any text data. To demonstrate the robustness to video compression, we embed the signature data into every frame of the host video sequence . LSB embedded video and its PSNR and MSE graph as shown in Figure5 and Figure6 and Table 2 shows MSE and PSNR values of parrot.avi video file.

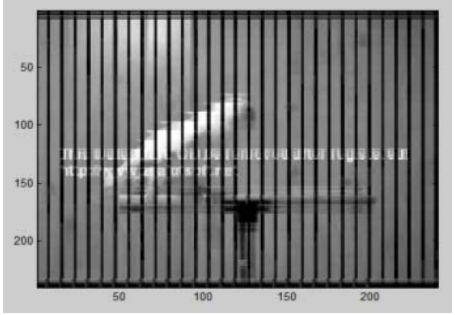

Figure 5: LSB Embedded video

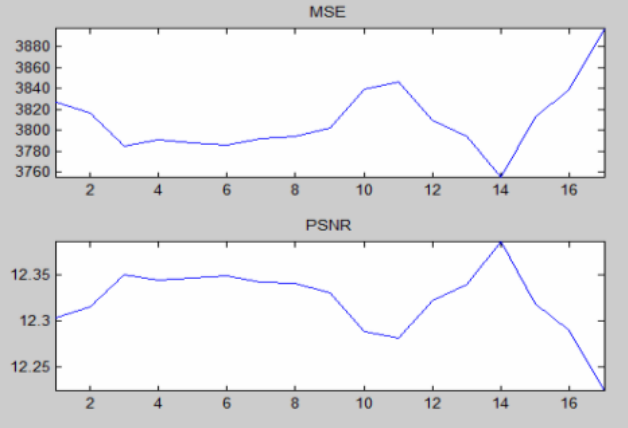

Figure 6: MSE and PSNR graph

Frame No	MSE	PSNR
1	3826.664323	12.302600
2	3816.337413	12.314336
3	3785.416111	12.349667
4	3791.091667	12.343161
5	3788.101858	12.346587
6	3786.357500	12.348587
7	3792.625868	12.341404
8	3793.800000	12.340059
9	3802.342986	12.330291
10	3839.556823	12.287993
11	3846.000799	12.280710
12	3809.719688	12.321873
13	3794.352448	12.339427
14	3754.816858	12.384916
15	3812.707049	12.318469
16	3838.762812	12.288891
17	3897.067899	12.223424

Table 1 : MSE and PSNR Values for video parrot.avi

VI CONCLUSION

In this paper a data hiding technique for high resolution video is proposed. The intension is to provide proper protection to data during transmission. For the accuracy of the correct message output that is extract from source we can use a tool for comparison and statistical analysis can be done. Its main advantage is that it is a blind scheme and its affect on video quality or coding efficiency is almost negligible. It is highly configurable, thus it may result in high data capacities. Here is 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 compression. Lossless recovery is important in embedding control or other binary data such as encrypted or encoded messages.

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] 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.

[3] 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.

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

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

[6] 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.

[7] J. Fridrich, M. Goljan, and R. Du. "Invertible Authentication Watermark for JPEG Images." ITCC 2001, Las Vegas, Nevada, April 2–4, 2001, pp.

[8] 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.

[9] "Fast multiplierless approximation of the DCT," in 33rd Annu. Conf. Information Sciences and Systems, Baltimore, MD, Mar. 1999,pp. 933–938.