SECURED DATA HIDING IN H.264 VIDEO USING INTRA PULSE CODE MODULATION

G.VADIVEL,P.VIGNESH,M.VIJAYA VENGATESH KUMAR, Mr. S.PREMKUMAR

NARASU'S SARATHY INSTITUTE OF TECHNOLOGY, SALEM

ABSTRACT: This paper proposes a novel scheme of reversible data hiding (RDH) in encrypted images using distributed source coding (DSC). After the original image is encrypted by the content owner using a stream cipher, the data-hider compresses a series of selected bits taken from the encrypted image to make room for the secret data. The selected bit series is Slepian-Wolf encoded using low density parity check (LDPC) codes Today internet plays an important role in data communication.Previously we have to send an message or information through telegram,letter,etc.But now it will be more easily to exchange information from one place to another through internet we can easily send also receive information.

1. INTRODUCTION

The widespread use of the Internet and World Wide Web has changed the way digital data is handled. The easy access of images, musical documents and movies has modified the development of data hiding, by placing emphasis on copyright protection, content-based authentication, tamper proofing, annotation and covert communication. Data hiding deals with the ability of embedding data into a digital cover with a minimum amount ofperceivable degradation, . Data hiding consists of two sets of data, namely the cover medium and the embedding data, which is called the message. The cover medium and the message can be text, audio, picture or video depending on the size of the message and the capacity of the cover.

2. INTRA MODE PREDICTION IN H.264/AVC

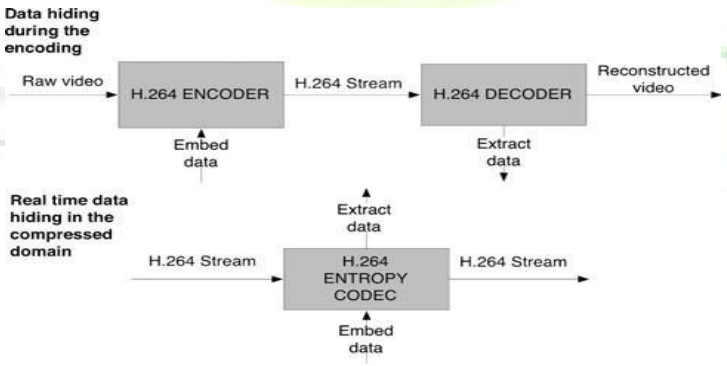

Fig. 3.1 The process of embedding and extracting data using the H.264 codec

All Rights Reserved @ 2016 IJARMATE

The H.264/AVC standard supports coded sequences containing I and P slices. I slices contain intra coded Fig. 1 The process of embedding and extracting data using the H.264 codecMacro blocks in which each 16 9 16 (I_16 9 16) or 4 9 4 (I_4 9 4) luma region and each 8 9 8 (I_8 9 8) chroma region is predicted from previously-coded samples in the same slice. A third type of Intra coding, called IPCM, is also provided for use in unusual situations. The encoder typically selects the prediction mode for each block that minimizes thedifference between the predicted block and the block to be encoded.

The I_4 9 4 mode is based on predicting each 4 9 4 luma block separately and is well suited for coding parts of a picture with significant detail. The I_16 9 16 mode, on the other hand, performs prediction and residual coding on the entire 16 9 16 luma block and is more suited for coding very smooth areas of a picture. In addition to these two types of luma prediction, a separate chroma prediction is conducted. In contrast to previous video coding standards (especially H.263 ? and MPEG-4 Visual), where intra prediction has been conducted in the transform domain, intra prediction in H.264/AVC is always con-ducted in the spatial domain, by referring to neighboring samples of previously-decoded blocks that are on the left and/or above the block to be predicted. Since this can result in spatio-temporal error propagation when inter prediction has been used for neighboring macro blocks, a constrained intra coding mode can alternatively be selected that allows prediction only from intra-coded neighboring macro blocks. In I_4 9 4 mode, each 4 9 4 lama block is predicted from spatially neighboring samples.

When the fidelity of the coded video is high (i.e., when the quantization step size is very small), it is possible in certain very rare instances of input picture content forthe encoding process to actually cause data expansion rather than compression. Furthermore, it is convenient for implementation reasons, to have a reasonably low identifiable limit on the number of bits necessary to process in a decoder in order to decode a single macro block. To address these issues, the standard includes an IPCM macro block

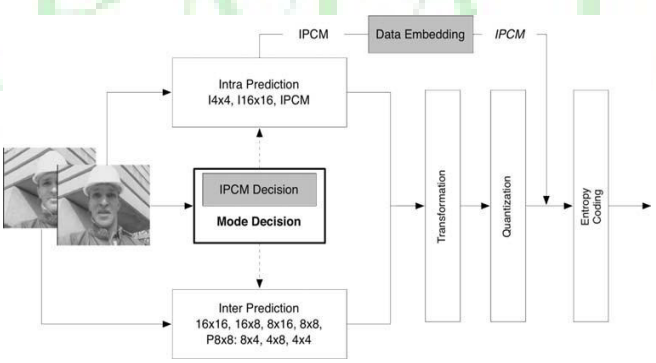

Fig. 3.2 Simplified block diagram of the proposed method integrated within the H.264 encoder

All Rights Reserved @ 2016 IJARMATE

3. THE PROPOSED DATA HIDING APPROACH

As mentioned in 'IPCM' is a macro block in which the values of the samples are sent directly without pre-diction, transformation, or quantization. The concept behind our method is to hide the desired data in the low bits of both the luma and the chroma samples of an IPCM macro block. Eventually, the hidden data will be embedded into the compressed H.264 stream intact. Simple and straightforward though, the proposed method has to face two practical obstacles: the rareness of the IPCM macro-blocks during the encoding andthe low efficiency of the IPCM mode in terms of compression. The latter turns out to be a trade off issue between the generated bitrate and the payload of the hidden data. This issue is discussed in Sect. with the help of some experimental results. Regarding the rareness of the IPCM macro block, we conducted several tests with many well-known video sequences. All of the tests resulted in none IPCM macro blocks no matter how low the quantization parameter was set. We therefore concluded that the only safe way to produce IPCM macro blocks during the encoding is to force the encoder to regard specific macro blocks as IPCM macro blocks. The simplified block diagram of the proposed method integrated within the H.264 encoder is shown in Fig.3.2.

According to the proposed method, two new steps are added in the H.264 encoder: the IPCM Decision and theData Embedding

The proposed method is characterized by three main features, namely, ease of implementation, respectable data capacity, and reusability. The latter allows data hiding in real time directly in the compressed domain. All these features are described below.

4. EASE OF IMPLEMENTATION

The proposed method can be easily integrated within the reference H.264 encoder. It takes place in a very early stage of the encoding process,before any spatial or temporal predictions and before the transformation and the quantization. Therefore, the impact of the proposed method to the encoding process is minimized. Some implementation hints on how to implement the proposed algorithm using the reference H.264 encoder version JM14.0 are given below:

Data capacity

The data capacity of a video sequence (YUV 4:2:0) is calculated in accordance with Eq. 1:

Data capacity ¼Luma capacity þ 2 _ Chroma capacity

where

Luma capacity ¼ 256 _ $N_{IPCM}\_L_{bits}$; and Chroma capacity ¼ 64 _ $N_{IPCM}\_C_{bits}$

N_{IPCM} is the number of the IPCM macro blocks used for data hiding, L_{bits} is the

All Rights Reserved @ 2016 IJARMATE

number of the low bits per IPCM luma sample used for data hiding, and C_{bits} is the number of the low bits per IPCM chroma sample used for data hiding. The luma and chroma samples are 256 and 64, respectively.

According to up to three low bits of an 8-bit sample can be modified without causing any visual distortion. However, our experiments showed that even if the four low bits are modified the distortion is imperceptible. This is explained by the fact that we are not dealing with static images but with moving frames at a rate of 30 fps. Moreover, weembed no more than one IPCM macro block per frame and not in successive frames. Finally, the rest of the non-IPCM macro blocks have possibly suffered greater distortion due to the intra/inter prediction and to the quantization during the encoding. In order to prove the above we zeroed the four low bits of every luma and chroma sample of the 49th macro block of the 9th frame of the mobile sequence. The sequence was encoded (QP = 28, CABAC) and decoded using the H.264 JM.14.0 codec. Figure 3.3 shows the visual result.

Fig. 3.3 Comparison of the visual results between the original and the marked 9th frame of mobile

An interesting approach would be the modifiable bits L_{bits} and C_{bits}tobe mathematically related to the Quantization Parameter (QP). For example for a high QP we could modify four bits while for a lower QP we could modify three bits or fewer. Other combinations are also applicable such as the use of three bits for the luma blocks and four bits for the chroma blocks. In the currentimplementation of the proposed method we modify the four low bits of both of the IPCM luma and the chroma samples in order to hide the data. Hence, from Eq. 1, a single IPCM macro block (N_{IPCM} = 1) for L_{bits} = C_{bits} = 4 gives a capacity of 1,536 bits. This might be regarded as the upper limit of the capacity per IPCM macro block.

5. SIMULATION RESULTS

The proposed algorithm was integrated within version JM14.0 of the reference H.264 software. The most important configuration parameters of the reference soft-ware are given in Table 3.1. The rest of the parameters have retained

All Rights Reserved @ 2016 IJARMATE

their default values.

The IPCM macro blocks are expected to have a negative impact to the produced bit rate. We conducted several tests in order to investigate this impact. For that purpose we used 300 frames or 10 s of well-known representative video sequences in QCIF format (YUV 4:2:0) such as the akiyo (Class A), the foreman (Class B) and the mobile (Class C). The QCIF format (176 9 144) was chosen because it is very common in mobile applications where the demand for real time is always high. The three classes that we used for our experiments possess the following characteristics:

Class A: Low spatial detail and low

amount of movement;
Class B: Medium spatial detail and low amount of movement or vice versa;
Class C: High spatial detail and medium amount of movement or vice versa

6 .H.264 PREDICTION AND MOTION ESTIMATION

Prediction is important layer in video codec and playing important role in compression and decompression. Prediction is used to remove spatial redundancy from individual image and temporal redundancy from moving images. Intra frame prediction and inter frame prediction are two types of prediction is used in video codec standards. this chapter will discuss theory and different algorithms for intra coding and inter coding.

7. MOTION COMPENSATION

Motion estimation and motion compensation are playing important role in the H.264 video codec. Motion compensation is considered as module of the predictionprocess. Motion estimation and motion compensation techniques are employed to removetemporal redundancy between consecutive frames of the video data. An encoded framehas enough amount of temporal redundancy after applying motion estimation. Theobservation indicates that either a camera or an object is moving in the moving picture.The difference between consecutive frames of video data should result into motion ofcamera or motion of object in frames.

The results of motion estimation are motionvectors, but this output does not match with desired output. Motion vector indicates thedisplacement of macro-block of the current frame in the reference frame. Motionestimation cannot remove temporal redundancy from consecutive frames of the videodata. The H.264 has introduced motion compensation block to remove complete

All Rights Reserved @ 2016 IJARMATE

temporalredundancy.Motion compensation regenerates a reference frame. The regenerated referenceframe is more likely the current frame, because the reference frame is generated using amotion vector of the current frame and a macro-block of the reference frame. Theregenerated reference frame is also known as a compensated frame. The difference of thecurrent frame and the compensated frame is known as a residual frame. The residualframe contains only a camera motion or an object motion in frame. Motion compensationtransmits the predicted error, which has much less correlation with the original frame.The predicted error can be coded at lower bit rate. The same prediction is generated at thereceiver side, and is combined with the received error to regenerate the current frame. Motion compensation enables low bit rates and low bandwidth features of H.264video codec.

The motion compensation can be implemented by fixed block size, variable blocksize, overlapped block size, and half pixel and quarter pixel. The current frame is dividedinto a number of non-overlapped macro-block. The motion vectors are estimated for allgenerated macro-blocks by motion estimation. The motion vectors are extra information,and require lots of memory on a storage space. In fixed block size, the current frame isencoded as conventional way. Thevariableblock sizeutilizes block-matchingcompensation, with ability to choose dynamic block size for video encoder. Thus, VBMC(variable block motion compensation) requires a less number of bits to represent a motionvector by merging the macro-blocks into a large block.

Due to non-overlapped blockmatching, the discontinuities are introduced at edges of the block. The overlapped blocksize avoids the discontinuities, which is induced in case of non-overlapped block size.The OBMC (overlapped block motion compensation) block size is twice big in eachdimension than non-overlapped block size. It is overlapping eight macro-blockssurrounded the original macro-block. The OBMC helps to improve quality of thereconstructed frame. The half-pixel and quarter-pixel motion compensation have abilityto calculate sub-pixel values. The sub-pixels are interpolated based on full-pixel valuesand full motion vectors by utilizing bi-cubic or bilinear 2D filtering. It treads off betweenhigh cost and complexity to accuracy. This chapter will discuss about FSBM (fixed sizeblock matching) and VSBM.

7.1 Fixed Block Motion Compensation

The fixed block motion compensation is the conventional technique for motioncompensation. The motion vectors and the reference frame inputs to FBMC module.FBMC generates the compensated current frame based on the given input, and additionalparameter such as a macro-block size, and a search area. The first motion vector isapplied as an input to FBMC. The FBMC moves into XY direction based on the motionvector. Then the macro-block is picked up based

All Rights Reserved @ 2016 IJARMATE

on XY-coordinates, and is placed intothe first location of a blank frame. The FBMC moves to the next motion vector. Thesesteps are repeated until FBMC walks through all the motion vectors. As a result, the compensated frame, and motion vectors without any modifications are outputted at theend of motion compensation. Figure 3.14 shows comparison of macro-block in FBMC andVBMC.

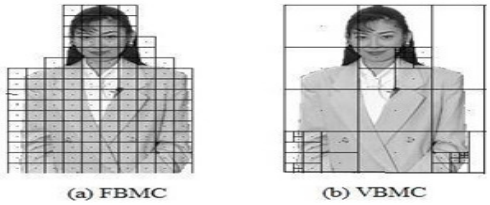

Figure 3.14The comparison of the macro-block size in FBMC and VBMC

8.CONCLUSION

The algorithm can preserve the bit-rate exactly even after encryption and data embedding, and is simple to implement as it is directly performed in the encrypted domain.Experimental results have shown that the proposed encryption and data embedding scheme can preserve file-size, whereas the degradation in video quality caused by data hiding is quite small.

REFERENCE:

1. Hong.W, Chen.T, and Wu.H (Apr. 2012), "An improved reversible data hiding in encrypted images using side match," IEEE Signal Process. Lett, vol. 19, no. 4, pp. 199–202.
2. Li.X.L, Yang.B, and Zeng.T.Y (Dec. 2011), "Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection," IEEE Trans. Image Process., vol. 20, no. 12, pp. 3524–3533.
3. Ma.K, Zhang.W, et al (2013). "Reversible Data Hiding in Encrypted Images by Reserving Room Before Encryption," IEEE Trans. Inf. Forensics Security, vol. 8, no. 3, 553-562.
4. W. Liu, W. Zeng, L. Dong, et al. "Efficient compression ofencryptedgrayscale images," IEEE Trans. on Image Processing,vol. 19, no. 4, pp. 1097-1102, 2010.
5. W. Liu, W. Zeng, L. Dong, et al. "Efficient compression of encrypted grayscale images," *IEEE Trans. on Image Processing*, vol. 19, no. 4, pp. 1097-1102, 2010

All Rights Reserved @ 2016 IJARMATE

IJARMATE

All Rights Reserved @ 2016 IJARMATE