Neurocomputing 188 (2016) 113–119

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

A new data hiding method for H.264 based on secret sharing

Yunxia Liu [a,b,*], Leiming Ju [c], Mingsheng Hu [a], Hongguo Zhao [b], Suimin Jia [a], Zhijuan Jia [a]

[a] Zhengzhou Normal University, Zhengzhou 450044, China
[b] Huazhong University of Science and Technology, Wuhan 430074, China
[c] Nanyang Institute of Technology, Nanyang, China

ARTICLE INFO

Article history:
Received 1 October 2014
Received in revised form
25 January 2015
Accepted 12 February 2015
Available online 17 December 2015

Keywords:
Data hiding
H.264/advanced video coding (AVC)
Secret sharing
Intra-frame distortion drift
Original video

ABSTRACT

This paper proposes a new data hiding method for H.264 without intra-frame distortion drift. The embedded secret data is first distributed into n sub-secrets with matrix equation by using Shamir's (t, n)-threshold secret sharing. Then we choose the block with prediction mode and embed the embedded data into discrete cosine transform (DCT) coefficients of the selected block. Finally, we recover the original video as much as possible when the hidden data is extracted out by using the t sub-secrets of the video frames. The experimental results show that this new robust data hiding method not only can get more robustness, effectively avert intra-frame distortion drift and get good visual quality, but also can protect the original video.

1. Introduction

In recent years, information security has gained significant attention due to the spread of illegal use of digital multimedia. Data hiding is an effective solution to embed the data in digital media for many applications such as law enforcement, copyright protection, user identification, and access control, etc. H.264 (H.264/AVC) is the most advanced standard for video compression with high compression efficiency [1]. It is well-adapted for network transmission. The existing information hiding methods for H.264, which embed data mainly in the quantized or un-quantized integer transform coefficients [2–6] and in the motion vectors [7], can be classified into two types according to their extraction approaches: detectable algorithm which inserts a code that can only be detected and readable algorithm which embeds a message that can be read. [2] and [3] belong to the first type, and [4–6],[8–11] belong to the latter type. Although hiding information into the digital video streams can be applied for a variety of applications, only the second type of algorithms can be used in covert communication and error concealment.

Intra-frame distortion drift is a big problem of data hiding in H.264 video streams, and the DCT coefficient of a block in an intra-frame technique is one of the most popular transform domain techniques and adopted in H.264. Data hiding for compressed media usually focuses on DCT domain since DCT is a widely used mechanism in edge-cutting compression standards such as JPEG and MPEG. A consensus in traditional algorithms for hiding data into images is "the less number of modification to the DCT coefficients, the less amount of distortion in the image" [9]. The existing methods have not handled the intra-frame distortion drift except [3], [8–11]. But [3] cannot extract the exact embedded content. The authors in [8] employ several paired-coefficients of a 4×4 DCT block for embedding data and compensating the intra-frame distortion. The algorithm presented in [9] employ several paired-coefficients of a 4×4 DCT block to accumulate the embedding induced distortion and use the directions of intra-frame prediction to avert the distortion drift. However, the algorithms in [8–10] do not consider the original video when the hidden data extracted out.

The robust data hiding algorithm in video data is proposed since the embedded bits sometimes cannot recover from network transmission, frame loss, video-processing operations, various attacks, and so on. All the algorithms in [4–6] embed the information into the DCT coefficients, and are more robust than the algorithms in [7–11]. Although the algorithms presented in [10,11] are compatible to H.264, they cannot resist the frame error such as frame loss, frame damage, etc. Secret Sharing has also been implemented in the literature [12,13], but it was not built on H.264. The infinite video sequences of H.264 just meet the redundancy properties of secret sharing. Consequently, further study and investigation are required.

In this paper, we present a data hiding algorithm that is robust, readable and can be used not only in video watermarking but also in covert communication and error concealment. The main

* Corresponding author.
E-mail address: liuyunxia0110@hust.edu.cn (Y. Liu).

http://dx.doi.org/10.1016/j.neucom.2015.02.102
0925-2312/

contributions of this work are as follows. First, we divide the original video into several groups by using the (t, n) secret sharing technique to improve the robustness of the embedded data. Second, we embed the embedded data into the coefficients of the 4×4 DCT block of the selected frames which meet our conditions to avert the distortion drift. Finally, we recover the original video as much as possible when the hidden data is extracted out. Then a new robust readable data hiding algorithm for H.264 without Intra-frame distortion drift is present.

The rest of this paper is organized as follows. Section 2 describes the proposed method and introduces the correlation theory of the method. Experimental results are presented in Section 3 and we draw the conclusions of this paper in Section 4.

2. The proposed method

In this section, we first introduce the correlation theory of the method, and then present the proposed method as well as the corresponding theoretical analysis in detail.

2.1. Embedding procedure analysis

The integer discrete cosine transform (ICT) which is developed from the DCT is used in H.264 standard. The transform based on 4×4 blocks is shown in (1). $W_{4\times4}$, which is the 'core' part of the transform, is the matrix of unscaled DCT coefficients corresponding to the residual block $Y_{4\times4}$.

$$W = C_f Y C_f^T \tag{1}$$

where

$$C_f = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{pmatrix}$$

The basic forward quantizer operation is given in (2) as follow:

$$\tilde{Y} = W \cdot round\left(\frac{PF}{Q_{step}}\right) \tag{2}$$

where Q_{step} is the quantizer step size which is determined by QP (QP is the quantization parameter) and PF is a^2,$\frac{ab}{2}$ or $\frac{b^2}{4}$ depending on the position (i, j) of the matrix PF as follows:

$$PF = \begin{pmatrix} a^2 & \frac{ab}{2} & a^2 & \frac{ab}{2} \\ \frac{ab}{2} & \frac{b^2}{4} & \frac{ab}{2} & \frac{b^2}{4} \\ a^2 & \frac{ab}{2} & a^2 & \frac{ab}{2} \\ \frac{ab}{2} & \frac{b^2}{4} & \frac{ab}{2} & \frac{b^2}{4} \end{pmatrix} \tag{3}$$

In order to simplify the arithmetic, the factor (PF/Q_{step}) is implemented in the reference model software as a multiplication by a factor MF and a right-shift, avoiding any division operations:

$$\tilde{Y} = round\left(W \cdot \left(\frac{MF}{2^{15+floor(\frac{QP}{6})}}\right)\right) \tag{4}$$

where

$$\frac{MF}{2^{15+floor(\frac{QP}{6})}} = \frac{PF}{Q_{step}}$$

At the decoder, after the re-scaling step as depicted in (5), the inverse ICT and the post-scaling step as described in (6), we can get the residual block R'.

$$W' = \tilde{Y} \cdot Q_{step} \cdot PF \cdot 64 \tag{5}$$

$$R' = round(C_i^T W' C_i / 64) \tag{6}$$

In data hiding algorithm, the encoded information is embedded into the quantized luminance DCT coefficients as in (7):

$$\tilde{Y}' = \tilde{Y} + \Delta \tag{7}$$

where $\Delta_{4\times4}$ is the error matrix added to the quantized DCT coefficient matrix $\tilde{Y}_{4\times4}$ by data hiding. $\Delta = (\alpha_{i,j})_{4\times4}$

After the re-scaling step as depicted in (8), the inverse ICT and the post-scaling step as described in (9) of the decoder, we can getR'' the residual block after embeddingΔ.

$$W' = \tilde{Y}' \cdot Q_{step} \cdot PF \cdot 64 = \tilde{Y} \cdot Q_{step} \cdot PF \cdot 64 + \Delta \cdot Q_{step} \cdot PF \cdot 64 \tag{8}$$

$$R'' = round(C_i^T W' C_i / 64) = round((C_i^T(\tilde{Y} \cdot Q_{step} \cdot PF)C_i) + C_i^T(\Delta \cdot Q_{step} \cdot PF)C_i) \tag{9}$$

The deviation of the pixel luminance value between the original block and the one after embedding is $E_{4\times4}$, where $E_{4\times4} = (e_{ij})_{4\times4}$, which can be calculated according to (10).

$$E = R'' - R' = round((C_i^T(\tilde{Y} \cdot Q_{step} \cdot PF)C_i) + C_i^T(\Delta \cdot Q_{step} \cdot PF)C_i) - round(C_i^T(\tilde{Y} \cdot Q_{step} \cdot PF)C_i) \tag{10}$$

Using B ($B = (b_{ij})_{4\times4}$) to express $C_i^T(\Delta \cdot Q_{step} \cdot PF)C_i$, we have

$$e_{i,j} = round(b_{i,j}) \text{ or } round(b_{i,j}) \mp 1 \tag{11}$$

Especially if $b_{i,j} = 0$, then $e_{i,j} = 0$.

2.2. Intra-frame prediction

A prediction block of H.264 intra prediction method is formed based on previously encoded adjacent blocks [1].The sixteen pixels in the 4×4 block (from a to p in Fig.1) are predicted by using the boundary pixels of the upper and left blocks which are previously obtained (from A to M), which use a prediction formula corresponding to the selected optimal prediction mode.

Each 4×4 block has nine prediction modes, four modes for each 16×16 luminance block. Figs. 2 and 3 describe these prediction modes respectively.

2.3. Intra-frame distortion drift prevention

The intra-frame distortion drift emerges because we embed bits into I frames. As illustrated in Fig. 4, we assume that current prediction block is $B_{i,j}$, then each sample of $B_{i,j}$ is the sum of the predicted value and the residual value, and the predicted value is calculated by using the samples which are gray in Fig. 4. Then the embedding induced errors in blocks $B_{i-1,j-1}$, $B_{i,j-1}$, $B_{i-1,j}$, and $B_{i-1,j+1}$ would propagate to $B_{i,j}$ because of using intra-frame prediction. This visual distortion that accumulates from the upper left to the lower right is defined as intra-frame distortion drift.

M	A	B	C	D	E	F	G	H
I	a	b	c	d				
J	e	f	g	h				
K	i	j	k	l				
L	m	n	o	p				

Fig. 1. Labeling of prediction samples [9].