用于内容认证的半脆弱可逆视频水印算法

曾 骁 陈真勇 范 围 陈 辉 熊 璋
(北京航空航天大学计算机学院, 北京 100191)

摘 要 提出了一种运用哈希函数进行 MPEG-4 内容认证的半脆弱可逆视频水印算法。在 YUV 视频进行 MPEG-4 编码的 I 帧中嵌入两个水印,一个运用哈希函数进行内容完整性验证并嵌入帧序号进行帧间篡改定位,另一个基于直流系数和低频系数用于帧内篡改定位。实验结果表明,算法能够对视频内容进行验证并实现篡改定位,且对 MPEG-4 压缩具有鲁棒性。另外,算法具有可逆性,能够在视频内容可信的前提下进行无损恢复。

关键词 视频水印 MPEG-4 半脆弱 可逆 篡改定位
中图法分类号: TP391 **文献标志码**: A **文章编号**: 1006-8961(2010)08-1189-07

Invertible Semi-fragile Video Watermarking Algorithm Used for Content Authentication

ZENG Xiao, CHEN Zhenyong, FAN Wei, CHEN Hui, XIONG Zhang
(*School of Computer Science and Engineering*, *Beihang University*, *Beijing* 100191)

Abstract This paper proposes an invertible semi-fragile video watermarking algorithm using a hash function to authenticate the MPEG-4 video contents. The proposed algorithm embeds two watermarks into I frame while encoding the YUV video to MPEG-4 format. One watermark using a hash function aims to authenticate the contents and embed the frame number for tamper localization between frames, and the other one based on direct coefficients and low-frequency coefficients is used for the detection of tamper localization within the frame. The experimental results show that the proposed algorithm is able to authenticate the video contents and detect the tamper localization, and it is robust for MPEG-4 compression. In addition, the proposed algorithm is exactly invertible, which means that the original video data is available as long as the watermarked video is credible.

Keywords video watermark, MPEG-4, semi-fragile, invertible, tamper localization

0 引 言

近年来,数字水印在鉴定数字媒体内容的完整性和真实性方面发挥着越来越重要的作用。用于认证的数字水印可分为脆弱水印和半脆弱水印两种[1]。脆弱水印对任何修改都是敏感的,适用于精确认证。半脆弱水印仅对一些诸如剪切、置换等恶意操作敏感而对诸如 JPEG,MPEG 有损压缩等常规处理具有鲁棒性,适用于内容完整性认证。在实际应用中,尤其在视频领域,为存储与传输的方便,一些压缩编码等处理是不可避免的,脆弱水印鲁棒性差,因此,其实用性远不如半脆弱水印。

传统的脆弱和半脆弱水印的嵌入都不具有可逆性,会对原始的宿主数据信息造成永久的改变。这在一些敏感的领域中,比如军事、法律、医学图像、政府机要等等是难以接受的,为此,可逆水印技术随之产生。可逆水印能够实现原始宿主信息的精确恢复,其思想最早由 Honsigner 等人在 1999 年的一项美国专利中提出[2],这项专利利用了可逆的嵌入方法在图像中嵌入数据,在提取嵌入数据的同时还能够完全恢复原始图像。

收稿日期:2008-09-11;**改回日期**:2009-04-22

第一作者简介:曾 骁(1981—),男。现为北京航空航天大学博士研究生。主要研究方向为信息隐藏和数字水印。
E-mail: zengxiao29@gmail.com

目前,尽管不少文献提出了可逆水印方案,但仅有很少一部分能用于半脆弱内容认证,且大多都是针对数字图像的。Fridrich 等人提出了两种用于 JPEG 图像认证的无损水印方法[3]。两种方法均采用了 LSB 嵌入机制和哈希函数方法。然而,此类方法对于许多合法处理操作如适当的 JPEG 压缩、噪声扰动等都比较敏感,显然不太适应实际应用场合。De Vleeschouwer 等人提出一种基于拼接理论的可逆半脆弱鉴定方案[4],它将图像分成不重复的块,并将每个块中的像素随机等分成两组,每组的像素值投射到一个圆上并计算出圆的重心,通过对两个圆重心的处理将一位数据嵌入到块中。通过实验验证了该方案对 JPEG 有损压缩具有鲁棒性,但是由于算法采用了模 256 加法(Modulo 256 addition)来防止上下限溢出,使得嵌入水印后的图像会遭受椒盐噪声的影响。

与数字图像水印相比,视频水印具有特殊性,其在图像水印所具有的特征基础上,还具有视频速率恒定性、与视频编码标准相结合以及实时性等特点。视频水印按照嵌入的时机可分为在原始视频中嵌入水印、编码压缩时嵌入水印和在压缩视频中嵌入水印。由于视频压缩的普遍性,对压缩视频水印的研究成为重点,其中在视频信息的 DCT 系数中嵌入水印是主流的水印算法。

目前对可逆视频水印的研究还比较少,Du 和 Fridrich 把 JPEG 图像中使用的可逆水印技术应用到 MPEG-2 视频中[5],但该算法是脆弱的。Shang 等人提出了一种用于 MPEG 压缩视频的可逆水印算法[6],其对 MPEG 压缩具有鲁棒性,采用通过选取合适的 DCT 系数对来提高水印的不可见性,并标记 DCT 系数的修改用于无损还原原始视频,但其仅用于信息隐藏,并不能进行内容认证。

视频内容认证的重要方面是对视频篡改的定位。视频篡改有别于图像篡改,Yin 在文献[7]中对其进行了分类:空间域篡改和时间域篡改,或者两者兼而有之。空间域篡改一般指对视频帧内容的篡改;时间域篡改建立在时间轴上,通常有添加、删除、重组或替换视频帧等处理。这些都是对视频进行篡改定位时应该考虑的因素。Park 等人对视频篡改问题进行了研究,提出了一种用于区分 MPEG-2 压缩和恶意篡改的可逆半脆弱水印算法[8],其将视频帧按 8×8 的块进行分割并作 DCT 变换,利用 DCT 变换直流系数的不变性将水印数据嵌入到 LSB 中。实验证明,其能够区别 MPEG-2 压缩和恶意篡改,并能够对帧内的篡改进行初步定位,但定位精度不高,对帧间的篡改无能为力。

可见,目前用于内容认证的半脆弱可逆水印的研究和应用比较少,篡改定位能力比较弱,与实际应用还存在一定的差距。并且研究绝大部分集中在图像上,可逆视频水印的研究尚处在初期阶段。但随着多媒体技术的发展和互联网的普及,视频的应用越来越广泛,对其内容完整性和真实性验证的重要性也越来越突出。在视频领域,亟需用于内容认证的半脆弱可逆水印算法。

本文提出了一种在视频 MPEG-4 压缩编码中进行可逆半脆弱水印的嵌入方法。在 I 帧嵌入能够篡改定位的可逆半脆弱水印,并能正确提取水印,精确还原原始视频。

1 MPEG-4 半脆弱可逆视频水印算法

在将原始 YUV 格式视频编码成 MPEG-4 视频流的过程中实施水印的嵌入,并且在提取水印后无损还原 MPEG-4 视频,即在检测端能使压缩视频完全恢复到未嵌入水印的标准压缩编码状态。

视频在进行 MPEG-4 压缩编码时,定义了 3 种类型的编码帧,分别为 I 帧(帧内编码帧)、P 帧(前向预测帧)和 B 帧(双向预测帧)。I 帧使用的是帧内编码,采用独立的编码方式实现压缩,不参考其他的帧信息,并且 I 帧为关键帧,不易被删除、跳过,因此在 I 帧中嵌入水印是常见的一种视频水印方法。如图 1 所示,在把原始 YUV 视频压缩到 MPEG-4 视频流的编码过程中,将水印信号嵌入到 I 帧 Y 通道(亮度通道)的量化 DCT 系数中。

1.1 水印嵌入过程

如图 2 所示,本文算法嵌入 2 个水印。第 1 个水印(左侧虚线框内)用于内容认证和帧间篡改定位,第 2 个(右侧虚线框内)用于帧内篡改定位。

第 1 个水印嵌入过程如下:

1) 原始 YUV 视频经 DCT 变换、量化后,得到 I 帧 Y 分量全部 8×8 块的量化 DCT 系数;

2) 计算全部量化 DCT 系数的 Hash 值。采用 SHA1 函数进行 Hash 计算,任何 DCT 系数的变化都会引起 Hash 值的改变,因而能够将此 Hash 值用于内容完整性验证;

3) 从每个 8×8DCT 块中选取一个量化系数,然后提取所有块这一系数的最小无效位(LSB)并组

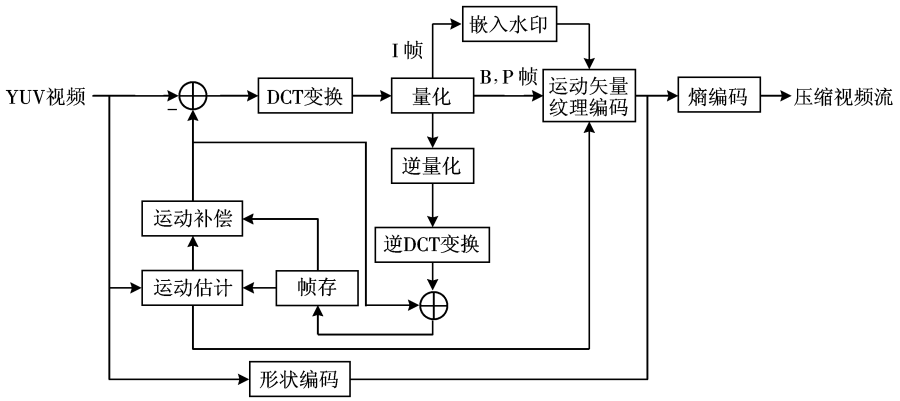

图 1　MPEG-4 编码及水印嵌入示意图

Fig. 1　Schematic diagram of MPEG-4 encoding and watermarking embedding process

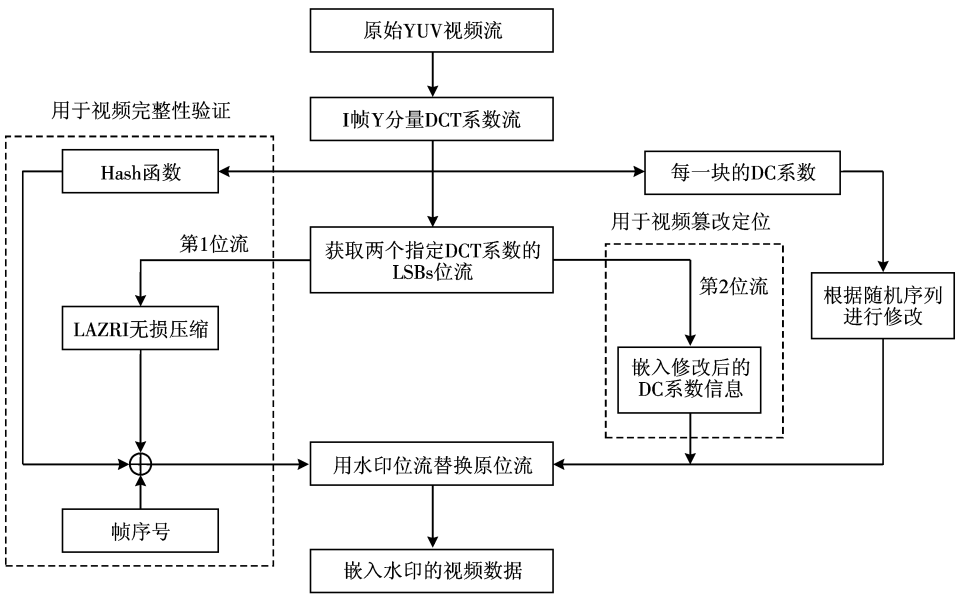

图 2　水印嵌入算法流程图

Fig. 2　Flowchart of watermarking embedding algorithm

成第 1 个 LSB 位流,将其进行 LAZRI 无损压缩以减小大小;

4) 将压缩后的 LSB 位流、Hash 值和帧序号连接起来,并用此组合位流替换选定系数的 LSB。从而完成第 1 个水印的嵌入。

从上述嵌入过程中可知,第 1 个水印信号包含了 Hash 值和帧序号,可以根据提取的水印信号中包含的 Hash 值验证内容的完整性,包含的帧序号判断添加、删除、重组或替换帧等操作以实现帧间篡改定位。并且,还可以根据水印信号中包含的被替换掉的 LSB 位流进行检测后的视频无损恢复。

第 2 个水印用于帧内篡改定位,本文算法基于每块的 DC 系数(直流分量)的如下特点进行设计:无论进行多少次压缩和解压缩以及无论采用何种量化表,其 DC 系数总保持不变。第 2 个水印嵌入可以分如下 3 步进行:

1) 从每个 8×8DCT 块中选取另一个量化系数并乘以 2,将该系数的 LSB 置 0。这样,可以往 LSB 位流中嵌入任何二元信息并且这些 LSB 位流能够被很容易恢复;

2) 将每一块的 DC 系数根据由密钥生成的随机序列进行更改。更改规则是,如果随机序列的该位为 1,则将对应的 DC 系数改为奇数,如果为 0 则改为偶数;

3) 如果块的 DC 系数有修改,则将该块选定系数的 LSB 置 1,反之置 0。

由第 2 个水印嵌入步骤可知,本文算法具有半脆弱性。因为嵌入水印后的视频无论再经过多少次压缩,DC 系数总保持不变,总是符合密钥和更改规则,这样水印就能够抵抗 MPEG-4 等压缩,从而具有半脆弱性。另外,算法采用密钥控制水印嵌入,使得算法具有安全性。

1.2 水印提取及检测过程

水印的提取是嵌入的逆过程,分为如下两步进行:

1)带水印的 MPEG-4 压缩视频的 I 帧经过熵解码后,得到 Y 分量 DCT 量化系数;

2)根据嵌入时选取的两个系数(每个水印对应一个系数),提取所有块对应系数的两个 LSB 位流(每个系数对应一个位流),也就是两个水印信息。其中第 1 个 LSB 位流包含了 Hash 值、帧序号和第 1 个水印嵌入的 DCT 系数的原 LSB 位流,第 2 个 LSB 位流包含了每个 8×8 块 DC 系数的修改信息。

根据提取的水印信息,能够对 MPEG-4 视频内容的完整性进行验证。验证分为帧间验证和帧内验证两部分。对于帧间的验证,根据提取的第 1 个水印中包含的帧序号能够判断每两个 I 帧之间是否存在添加、删除帧的情况。对于 I 帧内的验证,首先要根据 2 个水印信息对 I 帧进行还原,步骤如下:

1)运用第 1 个水印包含的原 LSB 位流信息进行第 1 个 DCT 系数 LSB 位的恢复;

2)根据第 2 个水印包含的修改信息对每块的 DC 系数进行恢复修正,并将第 2 个 DCT 系数的 LSB 对应置 0 后除以 2。

在以上的基础上再次计算出该帧的 Hash 值,利用该值与第 1 个水印中的 Hash 值进行比较进行水印检测。这里存在 3 种情况:

1)如果两个 Hash 值相等,证明视频是可信的并能无损恢复原始视频;

2)如果不相等,则检测量化 DC 系数。如果 DC 系数是正确按照由密钥生成的随机序列修改的话,则该视频是被有损压缩过;

3)如果 DC 系数的更改不符合随机序列,则说明被篡改过并且 DC 系数不符的块即是被有意篡改的块。

从以上水印提取与检测过程可知,本文算法嵌入的水印是能抵抗 MPEG 等压缩的半脆弱可逆视频水印,并且具有帧间和帧内定位篡改的能力。

1.3 篡改定位

引言中提到,篡改定位能力是视频内容认证的重要方面。并且,本文算法也给出了视频帧内和帧间篡改定位的方法。对于帧内篡改定位,本文算法采用密钥序列修改 DC 系数的方法进行篡改定位,但这种方法存在如下问题:算法仅对每个 8×8 块的 DC 系数的奇偶性进行修改并以此进行篡改定位。当篡改发生时,有可能存在对某 8×8 块 DC 系数进行了篡改但是并没有改变其奇偶性的情况,而这时算法并不会判定该块的 DC 系数被篡改过。这种不改变奇偶性的篡改情况总是以一定的概率发生,从而影响了算法篡改定位的精度。

为解决这一问题,采用如下思路对算法进行改进:仅靠对 DC 系数奇偶性的判断并不能保证足够高的篡改定位精度,如果在每个 8×8 块中多找几个系数并按照 DC 系数同样的处理方法进行水印嵌入,将会有效地降低定位的失误率,提高篡改定位精度。水印的嵌入和检测方法与上述算法的第 2 个水印相同。

在选取待判定的系数时考虑到:每帧视频图像的重要信息都集中在低频区域,对视频内容的篡改也多针对低频系数,因此应重点选取每块中的低频系数作为判定篡改的系数。但是,低频系数的修改对视频图像质量影响比较大,同时,每多嵌入一个用于篡改定位的系数信息也会对视频图像质量造成影响。本文将会在视频图像质量和篡改定位精度间寻找平衡并在第 2 节实验结果中给出相关讨论。

对于帧间篡改定位,采用在 I 帧中嵌入该帧序号的方法。这种方法通过对比实际帧序号和第 1 个水印中的帧序号,能够对两个 I 帧之间添加和删除帧的操作进行判断,但并不能准确定位增加或删除的帧,而且无法对 I 帧间的替换和重组帧的操作进行判断和定位。在此基础上,在视频的每一帧中都无损地嵌入帧序号,就能对上述帧间篡改进行定位。

2 实验结果及讨论

实验采用的视频测试序列为标准的 CIF 序列:akiyo(300 帧,352×288 像素)。对实验视频进行 4 种不同的水印嵌入流程,分别如下:

方式 1 按照第 1.1 节所述算法嵌入 2 个水印;

方式 2 在方式 1 的基础上,在第 2 个水印嵌

入中增加对 1 个系数的修改;

方式 3　在方式 1 的基础上,在第 2 个水印嵌入中增加对 2 个系数的修改;

方式 4　在方式 1 的基础上,在第 2 个水印嵌入中增加对 3 个系数的修改;

图 3(a)是实验视频的 I 帧图像,图 3(b)—图 3(e)是该帧按 4 种方式嵌入水印后的图像。从视觉上,由嵌入水印带来的失真完全不可见,并且采用 PSNR 值(峰值信噪比)来客观衡量图像失真。PSNR 计算公式如下:

$$\mathrm{PSNR} = 10\lg\frac{255^2}{\mathrm{MSE}} \tag{1}$$

式中,MSE 表示两幅图像的像素值均方差之和,MSE 的计算公式如下:

$$\mathrm{MSE} = \frac{1}{M \times N}\sum_{i=1}^{M}\sum_{j=1}^{N}(x_{i,j} - x'_{i,j})^2 \tag{2}$$

式中,M,N 分别表示图像长、宽方向的像素个数;$x_{i,j}$ 和 $x'_{i,j}$分别为 2 个图像中位于(i,j)坐标处的像素值。一般认为,图像的峰值信噪比在 36 dB 以上,人眼就基本上分辨不出图像的差异。本文算法的 PSNR 值如表 1 所示。

表 1　水印嵌入后 I 帧图像的 PSNR 值

Tab. 1　PSNR of the watermarking embedded I frames

	方式 1	方式 2	方式 3	方式 4
PSNR/dB	45.832 390	44.191 246	43.160 652	42.436 680

从表 1 能够看出,每种方式产生的压缩视频的 PSNR 值都在 40 以上,完全属于视觉可以接受的范围。而且由于在第 2 个水印中增加了对系数的修改,造成了 PSNR 值的降低,且每多修改一个系数,PSNR 值就有所减少。

PSNR 值只是图像差异的一个定量反映,并不能完全反映图像在人眼中的真实视觉感受,本文给出方式 1 ~ 4 产生的 akiyo 压缩视频中的某 I 帧截图进行对比。如图 3 所示,经仔细观察可以发现,方式 1 与原始 I 帧图像在视觉上几乎没有差别;从图 3(b) ~ 图 3(e)图像质量逐次下降,图 3(d)和图 3(e)已经能比较明显地看出背景和主题上的噪点,而 3(e)是这几幅图像中失真最严重的。

(a) 原始 I 帧图像

(b) 方式 1

(c) 方式 2

(d) 方式 3

(e) 方式 4

图 3　Akiyo 视频某 I 帧截图

Fig. 3　Capture of I frames in akiyo video

然后进行水印可逆性的验证。对提取到的水印与嵌入的水印进行比较,结果完全一致,表明本文算法中水印的嵌入和提取具有确定性。使用还原前后视频的 MSE 衡量本文算法的可逆性,结果如表 2 所示,表明最后得到的还原视频与标准 MPEG-4 编码输出的视频完全相同,从而验证了水印的可逆性。

表 2　还原后压缩视频与标准压缩视频的 MSE 值

Tab. 2　MSE of reduced compressed video and the standard one

	方式 1	方式 2	方式 3	方式 4
akiyo	MSE = 0	MSE = 0	MSE = 0	MSE = 0

对水印的半脆弱性进行验证。对嵌入水印的 MPEG-4 视频进行反复多次 MPEG-4 解压、压缩后，总能正确地提取水印。提取前后的 Hash 值不同，但 I 帧每块的 DC 系数总是遵循算法的修改规则，充分证明了水印对 MPEG-4 压缩具有鲁棒性。

最后，对水印的篡改定位能力进行验证。首先针对帧内的篡改定位进行讨论。在第 1.3 节中讨论了帧内篡改定位情况，认为采用对多个系数进行修改判定的方法有助于提高篡改定位能力。对上述 4 种水印嵌入方式进行篡改定位，结果如图 4 所示。

图 4(a)给出了方式 1 的某 I 帧篡改后的图，该 I 帧图像共受到了左上和右下方两处篡改，其中左上方的篡改为 15 个 8×8 块大小的区域，定为区域 1；右下方的篡改为 40 个 8×8 块大小的区域，为区域 2。图 4(b)—图 4(e)分别是方式 1—4 的篡改定位情况。表 3 给出了每个区域定位出的篡改块数(记为 N)和篡改检测率(记为 R)。区域篡改检测率 R 的计算公式为

$$R = \frac{N}{A} \tag{3}$$

式中，A 表示篡改区域的大小。

表 3 篡改定位结果

Tab. 3 Results of tamper localization

		方式 1	方式 2	方式 3	方式 4
区域 1	N	7	11	13	14
	R	0.467	0.733	0.867	0.933
区域 2	N	17	21	27	30
	R	0.425	0.525	0.675	0.750

(a) 方式 1 的 I 帧图像篡改

(b) 方式 1

(c) 方式 2

(d) 方式 3

(e) 方式 4

图 4 某 I 帧图像篡改定位结果

Fig. 4 Results of the tamper localization in I frames

通过对比容易发现，从方式 1～方式 4 篡改定位精度逐渐增大，篡改定位能力逐步增强。并且算法篡改定位准确度高，所有检测出的块均在篡改区域内。对比相关文献实验结果发现，方式 4 已经能够非常精确地进行篡改定位。

在帧间篡改定位中，对实验视频进行了随机帧插入和删除。通过提取第 1 个水印中的帧序号信息和当前 I 帧序号对比，能够判断出 2 个 I 帧间是否进行了帧的插入和删除操作。

从实验中的各种数据分析发现，本文算法能够对视频内容进行验证并能篡改定位，且对 MPEG-4 压缩具有鲁棒性。在确定水印视频是可信的情况下，还能对原始视频进行无损恢复。

3 结 论

本文算法通过在 YUV 视频进行 MPEG-4 编码的 I 帧中嵌入两个水印的方法，实现了对视频内容完整性的验证。以 MPEG-4 编码视频为例进行实验，实验结果表明，算法能够对两个 I 帧间增加、删除帧的操作进行判断，能对 I 帧图像内的篡改进行定位。并且能够抵抗 MPEG-4 有损压缩，具有半脆

弱性。本文算法还具有可逆性,在认定水印视频是可信的情况下能够对原始视频进行无损恢复。

参考文献(References)

[1] Cox I J, Miller M I. The first 50 years of electronic watermarking [J]. Journal of Applied Signal Processing, 2002(2): 126-132.

[2] Honsinger C W, Jones P W, Rabbani M, et al. Lossless Recovery of an Original Image Containing Embedded Data: US, 6278791[P]. 2001-08-21.

[3] Fridrich J, Goljan M, Du R. Invertible authentication [J]. Proceedings of SPIE, 2001, 4314: 197-208.

[4] Vleeschouwer C D, Delaigle J F, Macq B. Circular interpretation of bijective transformations in lossless watermarking for media asset management[J]. IEEE Transactions on Multimedia, 2003, 5(1): 97-105.

[5] Du R, Fridrich J. Lossless authentication of MPEG-2 video [C]//Proceedings of IEEE International Conference on Image Processing. Piscataway, NJ, USA: IEEE Press, 2002, 2: 893-896.

[6] Shang Y. A new invertible data hiding in compressed videos or images[C]//Proceedings of the 3rd International Conference on Natural Computation. Washington DC, USA: IEEE Computer Society, 2007: 576-580.

[7] Yin P, Yu H H. Classification of video tampering methods and countermeasures using digital watermarking[J]. Proceedings of SPIE, 2001, 4518: 239-246.

[8] Park J Y, Lim J H, Kim G S, et al. Invertible semi-fragile watermarking algorithm distinguishing MPEG-2 compression from malicious manipulation [C]//Proceedings of International Conference on Consumer Electronics. New York, NY, USA: IEEE Press, 2002: 18-19.