DOI: 10.16452/j.cnki.sdkjzk.2025.06.010 文章编号:1672-3767(2025)06-0110-12

H.264/AVC 域的自适应半脆弱视频水印算法

张长英,高梓航,辛珍瑶,范 迪
(山东科技大学 电子信息工程学院,山东 青岛 266590)

摘 要:为了更有效地保护视频信息的版权完整,实现视频帧和视频帧内容的篡改检测,本研究提出一种基于H.264/AVC域的自适应半脆弱视频水印算法。本算法结合非零量化系数和能量因子选择I帧中合适的子块,并对子块进行区域划分,构造视频特征序列、嵌入认证码和帧号,以保证算法对每个I帧中宏块的使用。算法的不可见性和鲁棒性实验结果表明,结构相似性指标在0.97以上,误比特率低于0.064 4。与同类算法相比,本算法在实现视频篡改检测的同时,具有良好的不可见性和抗重压缩性能。

关键词:视频水印;H.264/AVC;半脆弱;篡改检测;抗重压缩
中图分类号:TN929.5 **文献标志码**:A

A self-adaptive semi-fragile video watermarking algorithm based on H.264/AVC

ZHANG Changying, GAO Zihang, XIN Zhenyao, FAN Di
(College of Electronic and Information Engineering, Shandong University of Science and Technology, Qingdao 266590, China)

Abstract: In order to protect the copyright integrity of video information more effectively and achieve the tamper detection of video frames and video frame content, this paper proposes a self-adaptive semi-fragile video watermarking algorithm based on H.264/AVC. The algorithm combines non-zero quantization coefficients and energy factors to select suitable sub-blocks in the I frame and divides the sub-blocks into regions for constructing video feature sequences and embedding authentication codes and frame numbers to ensure the algorithm's use of macroblocks in each I frame. Experiments on the invisibility and robustness of the algorithm are conducted and the results, with a structural similarity index above 0.97 and a bit error rate below 0.064 4, show that the proposed algorithm has good invisibility and anti-recompression performance while achieving video tamper detection, and thus has obvious advantages compared to similar algorithms.

Key words: video watermarking; H.264/AVC; semi-fragility; tamper detection; anti-recompression

多媒体与互联网技术的飞速发展极大地方便了视频制作和传播,然而视频作品易于复制和篡改的特性也给信息安全与版权保护带来了严峻挑战。尽管数字水印技术可有效应用于版权保护,但现有视频水印算法仍难以兼顾视频篡改检测能力和鲁棒性。

按照作用域的不同,视频水印算法主要分为基于原始域水印和基于压缩域水印两类[1]。基于原始域的视频水印算法是在未经压缩编码的原始视频序列中完成水印嵌入及提取,常见方法包括空间域[2-3]和变换

收稿日期:2024-02-26
基金项目:国家语委"十三五"科研规划2019年度一般项目(信息化专项)(YB135-125);教育部产学合作协同育人项目(220900287094726)
作者简介:张长英(1999—),女,山东济宁人,硕士研究生,主要从事图像处理与分析方面的研究.
范 迪(1976—),女,河南南阳人,教授,博士,主要从事机器视觉、人工智能方面的研究,本文通信作者.
E-mail:fandi_93@126.com

域[4-7]等。相较而言,变换域算法复杂度较高,但通常能具有更好的性能。Fan 等[5]提出一种基于伪三维离散余弦变换(3-dimensional discrete cosine transform,3D-DCT)视频水印算法,结合伪下采样轮廓变换(non-subsampled contourlet transform,NSCT)、伪 3D-DCT 和非负矩阵分解(non-negative matrix factorization,NMF)等方法,取得了良好的鲁棒性。Farri 等[6]提出一种基于 Contourlet 变换和奇异值分解的盲视频水印算法,将水印嵌入低频子带。Sun 等[7]提出融合 NSCT、离散余弦变换(DCT)和舒尔(Schur)分解的视频水印技术。基于压缩域的视频水印算法则是在视频编解码过程中或在视频压缩后的码流中进行嵌入和提取,常用的视频压缩标准有 MPEG-X[8]、H.264/AVC[9-10]、H.265/HEVC[11-12]和 3D-HEVC[13-14]等。压缩域算法通常结合压缩标准所采用的变换方式来提升算法性能。Nguyen 等[15]使用熵编码器将原始视频流编码为帧内模式和量化离散余弦变换(quantification discrete cosine transform,QDCT)系数,通过构建 QDCT 系数的二维直方图实现水印嵌入。Fan 等[16]提出一种基于 H.264/AVC 的抗重压缩视频水印算法,将水印嵌入 DCT 量化系数中。

压缩域脆弱水印技术能够实现视频完整性认证和篡改定位,为保障当前视频作品的安全提供了有效手段。文献[17]提出一种基于能量关系的半脆弱视频水印算法,该算法采用 DCT 变换和能量计算,通过调整载波视频中间区域每个块反对角线上直流系数分量之间的关系完成水印嵌入。文献[18]提出一种基于奇异值分解(singular value decomposition,SVD)和离散小波变换(discrete wavelet transform,DWT)两种算法的视频内容半脆弱水印认证方案,通过加性嵌入算法先将认证码隐藏于小波中频子带中,再通过盲检测提取,但无法实现篡改定位功能。Zhang 等[19]提出一种基于 H.264/AVC 的视频半脆弱水印算法,利用最小代价函数在子块中选择对视频质量和比特率影响最小的 DCT 系数嵌入水印,从而显著降低对视频质量和比特率的影响,并且对恶意攻击表现出较高的敏感性。

综上,现有视频水印算法尚无法在实现视频篡改检测的同时保证视频水印的鲁棒性。为此,本研究提出一种基于 H.264/AVC 域的自适应半脆弱视频水印算法。该算法在视频 I 帧子块的 QDCT 系数中嵌入水印,对 I 帧中的子块进行区域划分,分别用于构造视频特征序列、嵌入认证码和嵌入帧号,从而充分利用每个 I 帧中宏块资源,实现以宏块为单位的篡改检测,并兼顾了算法的不可见性和鲁棒性。

1 半脆弱视频水印的嵌入算法

本研究设计的半脆弱视频水印算法基于 H.264/AVC 压缩标准,其编码器结构如图 1 所示[20]。H.264 编码的主要过程:首先计算当前帧与预测帧之间的残差,对其进行 DCT 变换和量化处理,然后对量化后得到的系数序列进行熵编码,最后写入码流。H.264 视频流可分为 I 帧、P 帧和 B 帧三种类型。其中,I 帧采用全帧压缩编码,不依赖其他帧进行预测;P 帧通过参考前序 I 帧和 P 帧进行预测编码;B 帧则依据前后参考帧进行双向预测编码。在 H.264 编码过程中,I 帧中的每个宏块被划分成 24 个 4×4 大小的子块,每个子块经 DCT 变换和量化后,得到 4×4 大小的 QDCT 系数矩阵。本研究选择在 I 帧的 QDCT 系数上嵌入水印。

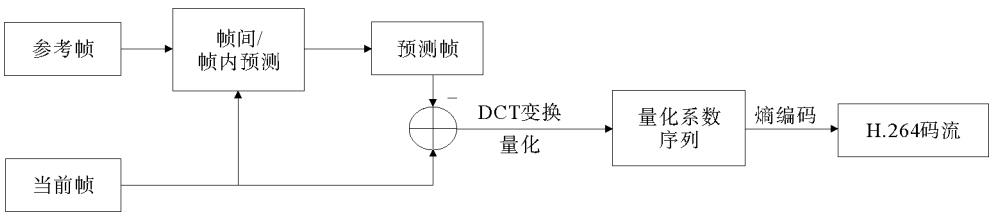


图 1 H.264/AVC 编码器结构图

Fig. 1 H.264/AVC encoder structure diagram

嵌入算法框架如图 2 所示。算法在 I 帧的每个宏块中选取一个子块,并划分为三个功能区,分别用于构造视频特征序列、嵌入认证码和嵌入帧号。为有效检测视频的丢帧、帧交换等篡改行为,并实现对篡改内容的定位,算法将当前帧号转换为二进制序列嵌入在相应 I 帧中。同时,将 I 帧中提取的视频特征矩阵与加密后的水印序列进行异或运算,生成认证码,作为半脆弱水印嵌入认证码嵌入区域。

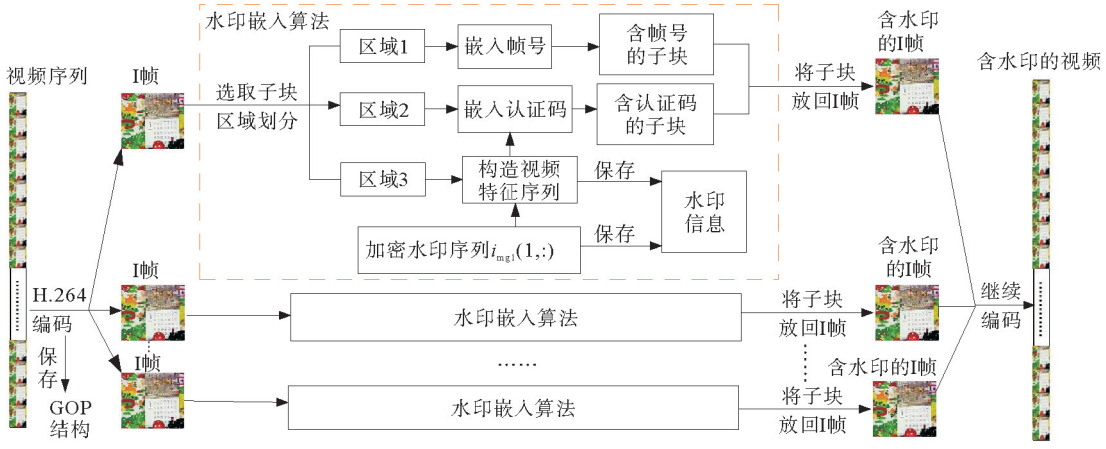
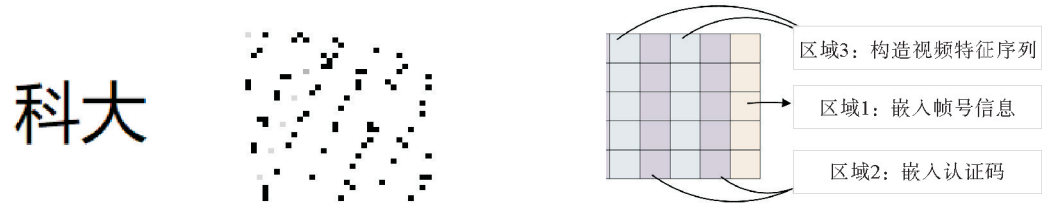


图 2 半脆弱视频水印的嵌入算法框架

Fig. 2 Embedding algorithm framework for semi-fragile video watermarks

1.1 半脆弱视频水印的生成算法

本研究以“科大”二字作为算法的水印信息,如图 3(a)所示。首先,对水印图像进行 Arnold 置乱加密,得到 32×32 大小的水印二值矩阵,如图 3(b)所示;然后将该二值矩阵转换为 1×1 024 大小的一维水印序列 $W(k)$ ($k=1,2,\cdots,1\ 024$)。本算法采用的半脆弱视频水印是结合视频内容与水印信息生成的:首先从视频帧中构造特征序列,然后与预处理后的水印序列进行异或运算,生成认证码。该认证码兼具视频帧信息和版权信息,可作为识别、定位视频帧的篡改和版权认定的依据。以 qcif 视频格式 I 帧为例,假设该帧包含 $M\times N$ 个宏块,从每个宏块中选择一个子块作代表,共同构成一个 $M\times N$ 子块阵列,如图 3(c)所示,其中每个方块代表一个子块,将该子块阵列划分为三个功能区:区域 1 用于嵌入帧号信息,区域 2 用于嵌入认证码,区域 3 用于构造视频特征序列。下面以第 i 个 I 帧为例,说明半脆弱视频水印的生成过程。



(a)原始水印图像 (b)置乱后水印图像 (c) I 帧子块区域划分

图 3 水印图像和 I 帧子块区域划分

Fig. 3 Watermark image and I frame sub-block region division

1) 选取第 i 个 I 帧中的子块。假设 I 帧共有 $M\times N$ 个宏块,在每个宏块中选取非零系数个数与能量因子均最大的一个子块,用于后续水印构造或嵌入,所有被选子块形成子块阵列,如图 3(c)所示。

2) 子块区域划分。对 $M\times N$ 个子块阵列区域划分为图 3(c)所示的区域 1、区域 2 和区域 3。区域 1 子块列记为 B_1;区域 2 与区域 3 的子块分别按列堆叠为一列,分别记为 B_2 和 B_3。

3) 为适应常见视频尺寸,本算法采用 32×32 大小的水印,根据实际视频尺寸,先将水印图像重置为 $P\times T$ 大小的水印图像,再进行 Arnold 置乱生成水印矩阵 $\boldsymbol{i}_{\mathrm{mgl}}$。其中,$P$ 为视频中 I 帧的个数,T 为 $M(N-1)/2$。

4) 构造视频特征序列。用区域 3 子块构造视频特征序列:

$$R_i(x,y)=\begin{cases}1, & Q_{x,y}^{[1]}\cdot Q_{x,y}^{[2]}>0;\\ 0, & Q_{x,y}^{[1]}\cdot Q_{x,y}^{[2]}<0。\end{cases}\tag{1}$$

式中:$x=1,2,\cdots,M$;$y=1,2,\cdots,(N-1)/2$,$R_i(x,y)$ 表示第 i 帧中位于第 x 行、第 y 列视频特征序列值,$Q_{x,y}^{[1]}$ 和 $Q_{x,y}^{[2]}$ 分别表示 B_3 子块阵列中第 x 行第 y 列子块的第一和第二个非零系数值。

5) 生成认证码。将视频特征序列 $R_i(x,y)$ 转换为 1 行 $M(N-1)/2$ 列,记为 $\boldsymbol{R}_i$,与加密水印序列进行异或运算,最终得到认证码,即半脆弱水印:

$$\boldsymbol{Z}_i=\boldsymbol{R}_i\oplus\boldsymbol{i}_{\mathrm{mgl}}(i,:)。\tag{2}$$

式中:$\boldsymbol{Z}_i$ 为第 i 个 I 帧的认证码,$\oplus$ 为异或操作,$\boldsymbol{R}_i$ 为第 i 个 I 帧的视频特征序列,$\boldsymbol{i}_{\mathrm{mgl}}(i,:)$ 为加密水印序列,每帧中的认证码共 $M(N-1)/2$ 位信息。

1.2 半脆弱视频水印的嵌入算法

本算法将半脆弱视频水印嵌入图 3(c)中区域 2 所对应的子块中,包括色度与亮度子块,在确保算法具备良好抗攻击能力的同时,支持以宏块为单位的篡改检测。同时,将帧号转换为二进制序列,嵌入图 3(c)中区域 1 子块中,可在检测端对帧的篡改进行检测。半脆弱视频水印嵌入算法与帧号嵌入方法一致,具体步骤如下。

1) 利用 H.264/AVC 编码器对原始视频进行编码,保存编码的 GOP(group of pictures)结构,并获得所有 I 帧。

2) 在第 i 个 I 帧的每个宏块中选择一个子块,记录其子块位置信息至数组 $\boldsymbol{D}_1$,并按图 3(c)所示将子块阵列划分为三个功能区。

3)将帧号转化为 b 位二进制序列 $F_i(v)$,$v=1,2,\cdots,b$;b 由 P 转化为二进制序列的位数决定。调制每个子块中第 4 位非零系数,将 $F_i(v)$ 按位嵌入到区域 1 的子块中,调制方式为:

$$Q_v^{[4]}=\begin{cases}-Q_v^{[4]}, & \mathrm{mod}(p(v),2)\neq F_i(v);\\ Q_v^{[4]}, & \mathrm{mod}(p(v),2)=F_i(v)。\end{cases}\tag{3}$$

式中:$Q_v^{[4]}$ 为嵌入帧号区域 B_1 中第 v 个子块的第 4 位非零量化系数;mod( )为取余运算,结果为 0 或 1;$p(v)$为第 v 个子块中第 4 和第 5 位非零量化系数中正数的个数;$F_i(v)$为第 v 位二进制序列。

4) 区域 B_2 子块阵的大小为 M 行$(N-1)/2$ 列,将 B_2 子块阵列转换为 1 行 $M(N-1)/2$ 列,记为 B'_2,调制 B'_2 中第 4 和第 5 位非零量化系数中正数个数实现认证码 $\boldsymbol{Z}_i$ 的嵌入,每帧共嵌入 $M(N-1)/2$ 位的认证码信息,调制方式为:

$$Q_x^{[4]}=\begin{cases}-Q_x^{[4]}, & \mathrm{mod}(p(x),2)\neq Z_i(x);\\ Q_x^{[4]}, & \mathrm{mod}(p(x),2)=Z_i(x)。\end{cases}\tag{4}$$

式中:$Q_x^{[4]}$ 为认证码嵌入 B'_2 中第 x 个子块的第 4 位非零量化系数,$x=1,2,\cdots,M(N-1)/2$;$p(x)$ 为第 x 个子块中第 4 和第 5 位非零量化系数中正数的个数;$Z_i(x)$ 为第 x 位认证码。

5) 将嵌入水印和帧号的子块放回原位置,与其他未修改子块共同生成含水印的 I 帧。

6) 重复步骤 2)~5),直到视频中所有 I 帧全部嵌入认证码和帧号后,将 32×32 大小的水印矩阵、子块位置和认证码作为密钥保存至第三方,继续视频编码得到含水印的 H.264 视频码流,完成全部水印嵌入获得含水印视频。半脆弱视频水印嵌入算法的伪代码如算法 1。

算法 1 半脆弱视频水印的嵌入算法

输 入:视频序列 V,水印图像 $\boldsymbol{W}_1$;

初始化:$\boldsymbol{W}_1=\text{zeros}(32,32)$,$\boldsymbol{W}=\text{zeros}(P,T)$;

开 始:使用 H.264 编码对 V 进行编码;

在所有 I 帧的每个宏块 B_m 中选择一个子块,将子块位置记录在数组 $\boldsymbol{D}_1$ 中,共 P 个 $M\times N$ 大小的子块,记为 B_k;

/* P 为视频中 I 帧的个数 */

$\boldsymbol{W}=\text{reshape}(\boldsymbol{W}_1,P,T)$ /* $T=N(M-1)/2$ */

$\boldsymbol{i}_{\text{mgl}}=\text{Arnold}(\boldsymbol{W})$

对第 i 个 I 帧的子块 B_{ki} 划分区域。区域 1 用于嵌入帧号,区域 2 用于嵌入认证码,区域 3 用于构造视频特征序列

对区域 3,根据第 1 和第 2 个非零系数值构造视频特征序列 $\boldsymbol{R}_i$;

计算认证码 $\boldsymbol{Z}_i$; /* $\boldsymbol{Z}_i=\boldsymbol{R}_i\oplus\boldsymbol{i}_{\text{mgl}}(i,:)$ */

对区域 2,调制第 4 和第 5 位非零量化系数中正数个数实现认证码 $\boldsymbol{Z}_i$ 的嵌入;

将第 i 帧的帧号转为第 v 位二进制序列 $F_i(v)$; /* v 为视频的总帧数 */

对区域 1,调制第 4 位非零系数的正数个数来实现帧号 F_i 的嵌入,得 B'_{ki};

把 B'_{ki} 放回 I 帧;

将视频中所有的嵌入认证码和帧号的 I 帧放回 V;

继续使用 H.264 编码对 V 进行编码;

输 出:含水印视频 V_1,子块位置 $\boldsymbol{D}_1$,构造的认证码 $\boldsymbol{Z}_i$,水印矩阵 $\boldsymbol{i}_{\text{mgl}}$。

2 半脆弱视频水印的提取及篡改检测算法

半脆弱视频水印的提取及篡改检测算法在 H.264 视频解码过程中完成。H.264 视频解码器结构如图 4 所示。解码器首先从码流中获取数据信息,然后通过熵解码与重排序后得到变换系数,再对其进行反量化和反 DCT 变换,从而获得残差块。同时,解码器利用从码流中解析出的头信息进行帧内预测,生成预测块。将预测块与残差块相加,获得帧图像,最后经过环路滤波处理,输出重建帧。一个图像帧层中有 I 帧、P 帧和 B 帧等不同类型,本研究的水印提取操作针对 I 帧进行。

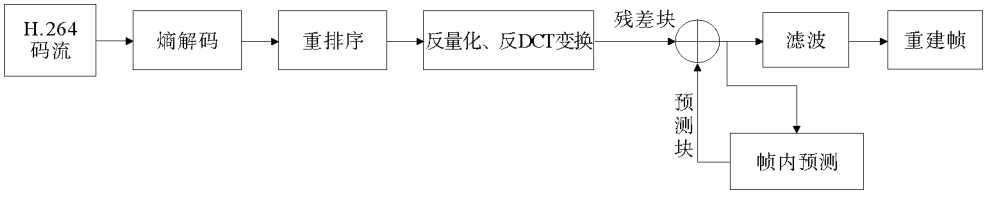


图 4 H.264/AVC 解码器结构图

Fig. 4 H.264/AVC decoder structure diagram

2.1 半脆弱水印提取流程

半脆弱水印提取算法包括帧号提取、水印图像提取以及视频篡改检测。水印提取过程如图 5 所示,先从待检测视频序列中提取视频特征序列,再将其与从码流中解析出的认证码进行异或运算,从而还原水印图像。最后,分析所提取的水印图像与认证码,实现对视频内容篡改的检测。水印提取步骤如下。

1) 按照保存的 GOP 结构,使用 H.264/AVC 编解码器对视频进行解码,得到待检测视频中所有的 I 帧,根据数组 $\boldsymbol{D}_1$ 提取出每个宏块中含水印信息的子块。

2) 对第 i 个 I 帧中的子块进行区域划分。

3) 将嵌入认证码区域中的子块阵转换为 1 行 $M(N-1)/2$ 列,对每个子块的第 4 和第 5 位非零量化系数的正负情况进行分析,提取认证码。提取算式为:

$$Z'_i(x)=\begin{cases}1, & \text{mod}(p(x),2)=1;\\ 0, & \text{mod}(p(x),2)=0。\end{cases} \tag{5}$$

式中：$Z'_i(x)$ 为对第 i 个 I 帧提取到的第 x 位认证码，其中 $x=1,2,\cdots,M(N-1)/2$；$p(x)$ 为认证码嵌入区域第 x 个子块中第 4 和第 5 位非零系数中正数的个数。

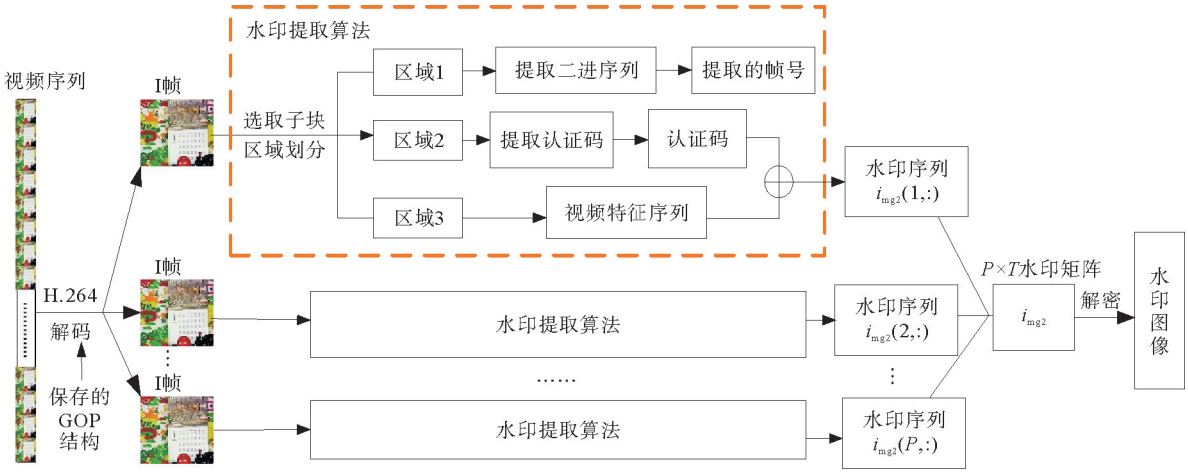


图 5　水印提取过程

Fig. 5　Watermark extraction process

4）提取嵌入帧号区域的 b 位二进制序列，如式(6)所示，将二进制序列转化为十进制，得到提取的帧号，

$$F'_i(v)=\begin{cases}1, & \mathrm{mod}(p(v),2)=1;\\ 0, & \mathrm{mod}(p(v),2)=0。\end{cases} \tag{6}$$

式中：$F'_i(v)$ 为提取的第 i 个 I 帧的二进制帧号序列，$p(v)$ 为帧号嵌入区域第 v 个子块中第 4 和第 5 位非零系数中正数的个数。由于保存的 $P\times T$ 水印矩阵中 P 为原视频中 I 帧的个数，因此 v 的最大值由 P 转为二进制的位数决定。

5）在构造视频特征矩阵区域，将当前 I 帧的视频特征序列与提取的认证码进行异或运算，得到第 i 个提取的水印序列 $\boldsymbol{i}_{mg2}(i,:,:)$。

6）重复步骤 2)～5)，得到提取的水印矩阵 $\boldsymbol{i}_{mg2}$，将水印矩阵进行重置和反置乱解密，最后得到提取的 32×32 大小的水印图像。半脆弱水印提取算法的伪代码如算法 2 所示。

算法 2　半脆弱视频水印提取算法

输　入：待检测视频 V_2，子块位置 $\boldsymbol{D}_1$；

初始化：$\boldsymbol{W}_1=\mathrm{zeros}(32,32)$，$\boldsymbol{W}_2=\mathrm{zeros}(P,T)$；

开　始：使用 H.264 解码对 V_2 进行解码；

　　提取 V_2 中的所有 I 帧；

　　根据数组 $\boldsymbol{D}_1$ 提取每个宏块中含水印信息的子块，共 P 个 $M\times N$ 大小的子块；　　/* P 为视频中 I 帧的个数 */

　　对第 i 个 I 帧中的子块划分区域；

　　根据区域 1 中每个子块的第 4 和第 5 位非零系数中正数的个数提取第 i 个 I 帧的帧号 F'_i；

　　对区域 2 中每个子块的第 4 和第 5 位非零系数的正负情况进行分析，提取认证码 $\boldsymbol{Z}'_i$；

　　对区域 3 中每个子块的第 1 和第 2 位非零系数值的正负情况进行分析，提取视频特征序列 $\boldsymbol{R}'_i$；

　　将提取的视频特征序列 $\boldsymbol{R}'_i$ 与提取的认证码 $\boldsymbol{Z}'_i$ 进行异或运算，得到第 i 个提取的水印序列 $\boldsymbol{i}_{mg2}(i,:,:)$；　/* $\boldsymbol{i}_{mg2}(i,:,:)=\boldsymbol{R}'_i\oplus\boldsymbol{Z}'_i$ */

　　对所有 I 帧提取出对应的水印序列，得到提取的水印矩阵 $\boldsymbol{i}_{mg2}$；

　　$\boldsymbol{W}_2=\mathrm{IArnold}(\boldsymbol{i}_{mg2})$；　　/* IArnold( )为 Arnold 变换的逆变换 */

　　$\boldsymbol{W}_1=\mathrm{reshape}(\boldsymbol{W}_2,32,32)$；

输　出：提取的水印图像 $\boldsymbol{W}_1$。

2.2 视频篡改检测流程

本算法在视频防篡改检测设计中,将帧号嵌入对应的视频帧,可初步识别视频的丢帧、帧交换等攻击行为;认证码的设计能进一步识别并定位视频帧内容的篡改。基于提取的水印图像与认证码,视频篡改检测流程如下。

1) 根据保存的 $P \times T$ 水印矩阵可获取原视频的 I 帧数为 P ,若待检测视频中的 I 帧个数 $p < P$,则判定视频遭受丢帧攻击;若 $p > P$,则判定视频遭受帧添加攻击。通过分析每一帧的帧号顺序,可定位丢失或增加的视频帧位置。

2) 若待检测视频中的 I 帧个数 $p = P$,则进一步判断提取的帧号与视频帧的实际位置是否一致。若一致,说明嵌入帧号区域均未被篡改;若不一致,则将提取到的二进制序列与当前视频帧号的二进制序列对比,若第 v 位序列值错误,表明嵌入帧号区域的第 v 个宏块遭到篡改。

3) 将提取的水印矩阵 $\boldsymbol{i}_{\mathrm{mg2}}$ 与原始水印矩阵 $\boldsymbol{i}_{\mathrm{mg1}}$ 进行对比,若发现不一致的水印序列 $\boldsymbol{i}_{\mathrm{mg2}}(j,:)$,则可判定第 j 帧中的构造视频特征序列区域或嵌入认证码区域存在篡改。

4) 若水印序列 $\boldsymbol{i}_{\mathrm{mg2}}(j,:)$中的错误水印信息为 $\boldsymbol{i}_{\mathrm{mg2}}(j,x)$,说明是构造视频特征序列区域或嵌入认证码区域中第 x 个宏块被篡改。在此情况下,进一步将保存的第 j 帧的认证码与提取的认证码进行比对。若第 x 个认证码正确,证明提取认证码的子块无错误,篡改发生在构造视频特征序列区域第 x 个宏块。若第 x 个认证码错误,则提取第 x 位视频特征序列值与原始视频特征序列值进行比较。若二者相等,说明篡改位于嵌入认证码区域的第 x 个宏块;若不等,则篡改位于构造视频特征序列区域第 x 个宏块。原始视频特征序列值可通过保存的认证码与水印矩阵异或运算得到。

3 实验结果与分析

本算法在 H.264/AVC 参考编码软件 JM8.6 和 MATLAB 2016b 平台实现。为验证算法有效性,实验选取了涵盖多种场景的测试序列,包括 Hall.qcif、Mobile.qcif、Foreman.qcif、News.qcif、Akiyo.cif 等视频,每段视频均为 300 帧,并以“科大”作为嵌入的水印图像。6 个视频的实验参数具体设置如表 1 所示,除视频尺寸外,帧率等其他参数值均一致。其中,Akiyo.cif 的尺寸大小为 352×288 像素,其他视频的尺寸大小为 176×144 像素。本节主要从不可见性鲁棒性及篡改检测能力等方面评估水印方案的性能。

表 1 实验参数配置

Table 1 Experimental parameter configuration

参数类型	参数值
帧率	30 帧/s
GOP 结构	IPPP
编码档次	Baseline
熵编码类型	CACVL
编码帧数	300
初始量化参数	20

3.1 不可见性实验

为了验证本算法的不可见性,将半脆弱水印嵌入视频序列的各帧中,然后分别计算 6 个测试视频在嵌入水印后 300 帧的结构相似性(structural similarity index,SSIM)均值、峰值信噪比(peak signal to noise ratio,PSNR)均值以及提取出水印的归一化相关系数(normalized correlation,NC)值[21],实验结果如表 2 所示。在未遭受恶意攻击的情况下,各视频的 SSIM 均值均高于 0.97,说明水印对视频质量影响较小,算法具备良好的不可见性;提取水印的 NC 值均为 1,表明在无攻击条件下能够完整、准确地提取水印信息。

表 2 不可见性实验结果

Table 2 Invisibility experiment results

实验视频	PSNR 均值/dB	SSIM 均值	无攻击下水印 NC 值	无攻击下帧号提取
Foreman	39.569 2	0.987 4	1.000 0	True
News	38.728 2	0.986 7	1.000 0	True
Mobile	40.916 4	0.997 1	1.000 0	True
Hall	35.816 3	0.973 3	1.000 0	True
Akiyo	40.539 0	0.991 8	1.000 0	True
Container	37.948 5	0.993 6	1.000 0	True

3.2 鲁棒性实验

3.2.1 抗重压缩测试

为检验本算法的抗重压缩能力,对 6 个已嵌入水印的视频进行等量化值参数(quantization parameter,QP)重压缩实验(原始 QP 为 20),并计算重压缩后水印图像的 NC 值和误比特率(bit error rate,BER)[22]值,实验结果如表 3 所示。可以看出,重压缩后提取出的水印图像的 NC 值大于 0.97,BER 均小于 0.02,表明本算法具有良好的抗重压缩能力。

表 3 重压缩实验结果

Table 3 Recompression experiment results

实验视频	Foreman	News	Mobile	Hall	Akiyo	Container
BER	0.010 0	0.014 4	0.001 8	0.007 4	0.003 3	0.010 0
NC	0.979 7	0.988 4	0.995 7	0.972 8	0.989 6	0.983 6

为进一步评估抗重压缩性能,在第一次重压缩的基础上对实验视频进行第二次重压缩。如图 6 所示,即使经历两次重压缩,提取水印的 NC 值仍保持在 0.96 以上,且 BER 值均小于 0.02,能够有效保障视频版权,表明本算法对重压缩攻击具有较好的鲁棒性。

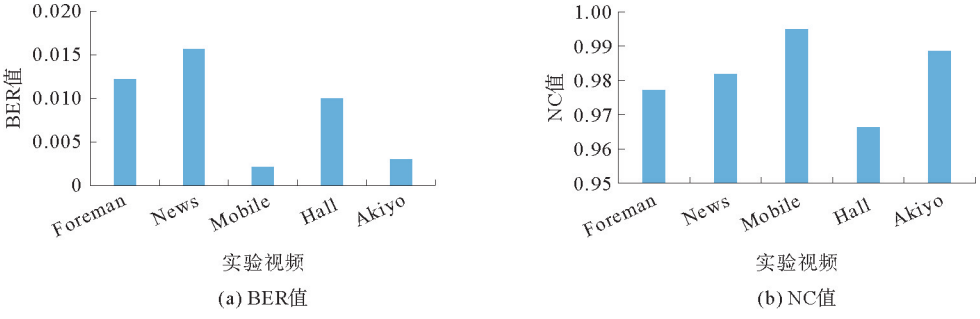


图 6 第二次重压缩实验结果

Fig. 6 Results of the second recompression experiment

3.2.2 抗重量化测试

本算法虽然是半脆弱视频水印,但在设计阶段已考虑对重量化攻击的抵抗能力。为验证其抗重量化性能,在初始 QP 为 20 时进行视频编码,随后调整 QP 为 18～22 进行重量化处理,提取重量化后的水印图像,并计算 BER 值,结果如表 4 所示。由表 4 可知,重量化后水印提取的错误率均低于 0.064 4,说明算法在遭受重量化攻击时仍能有效提取水印,具备良好的抗重量化能力。

表 4 半脆弱水印在重量化攻击下提取的 BER 值

Table 4 BRE value of semi-fragile watermark under requantization attack

实验视频	QP 值				
	18	19	20	21	22
Foreman	0.053 3	0.028 9	0.010 0	0.042 2	0.064 4
News	0.022 2	0.024 4	0.014 4	0.020 0	0.042 2
Mobile	0.055 6	0.040 0	0.018 0	0.040 0	0.057 8
Hall	0.033 3	0.022 2	0.007 4	0.031 1	0.042 2
Akiyo	0.037 8	0.020 0	0.003 3	0.040 0	0.044 4

3.3 篡改检测实验

在视频防篡改检测方案中,本算法将帧号嵌入对应的视频帧内,可初步识别如丢帧、帧添加、帧交换等攻击行为,并通过检测帧号是否异常来确认是否存在篡改。此外,认证码的设计进一步支持对视频帧内容的识别与篡改定位。

1) 丢帧检测实验。以 Foreman 视频为测试对象,对其第 60～80 帧实施丢帧攻击,然后识别待检测视频

的帧号,检测结果如图 7 所示。实验结果表明,本算法能够准确提取前 1～60 帧的正确帧号,并在丢帧位置识别出帧号错误,有效检测出丢帧篡改。

2) 帧添加检测实验。为验证算法对帧添加攻击的检测能力,在视频第 120 帧后插入原序列中的第 280～300 帧。帧号提取结果如图 7,可以看出,前 120 帧的帧号提取结果是正确的,第 121～140 帧所提取的帧号为 280～300,表明本算法能够有效识别帧添加攻击。

3) 帧交换检测实验。先将视频中第 80～120 帧与第 260～300 帧进行位置交换,再提取视频的帧号。从图 7 可以看出,在第 80 帧处提取到帧号为 260,而在第 260 帧处提取到帧号为 80,说明本算法能够准确检测出因帧交换导致的视频结构篡改。

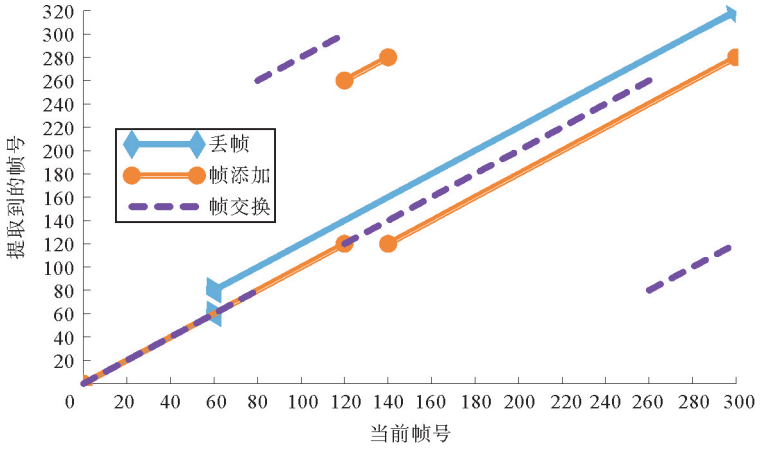
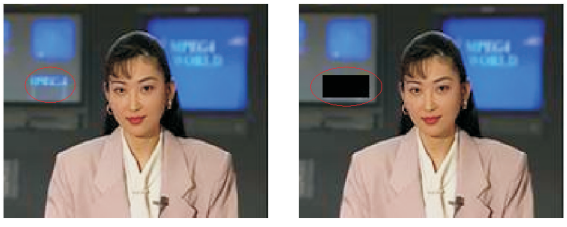


图 7 丢帧、帧添加、帧交换攻击下帧号提取结果

Fig. 7 Frame number extraction results under frame loss, frame addition, and frame exchange attacks

4) 内容篡改定位实验。为测试算法对视频帧内容篡改的定位能力,对 Akiyo 视频第一帧中部分区域进行篡改,通过本算法在视频帧中标记出被篡改宏块位置,如图 8 中黑色方块所示。从图 8 可以看出,本算法能够以宏块为单位,定位到视频帧内容被篡改的位置。

(a) 原始图像 (b) 遭篡改图像1 (c) 对篡改区域的定位1



(d) 遭篡改图像2 (e) 对篡改区域的定位2

图 8 Akiyo 视频帧篡改检测

Fig. 8 Akiyo video frame tampering detection

3.4 与同类算法对比实验

为全面评估算法性能,本研究与文献[23-24]的方法进行了对比,所有实验参数均与原文献保持一致。文献[23]通过对色度子块预测模式进行分组,设计认证码,并通过修改色度子块中三个中频系数之间的关系嵌入水印;文献[24]将视频帧数编码为二进制序列,并将其作为水印嵌入当前帧,并基于倒数第二与倒数第三个非零系数之间的数值关系构造认证码。

3.4.1 不可见性比较

本研究比较了嵌入水印前后视频帧的 SSIM 值,并与文献[23]、文献[24]的结果进行对比,实验结果如图 9 所示。由于文献[24]未使用 Akiyo 数据进行不可见性实验,故图中未包含该数据。从图 9 可以看出,本算法在所有测试序列中的 SSIM 值均高于对比文献,表明其具有更好的不可见性。

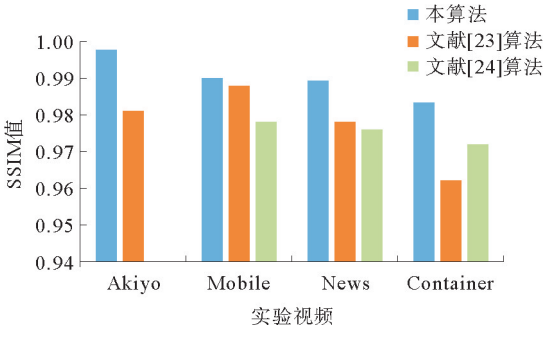


图 9 本算法与对比文献的 SSIM 值比较

Fig. 9 Comparison of SSIM values between our algorithm and comparative literature

3.4.2 抗重压缩和重量化比较

文献[23]在重压缩实验中采用的 QP 值为 28。为对比公平,本算法在相同 QP 值下进行重压缩实验,对比结果如图 10 所示。可以看出,除 Container 视频的水印 NC 值略低于文献[23]外,本算法在其他测试视频中的 NC 值均高于对比文献,而 BER 值均小于对比文献,表明本算法在抗重压缩性能方面总体更优。

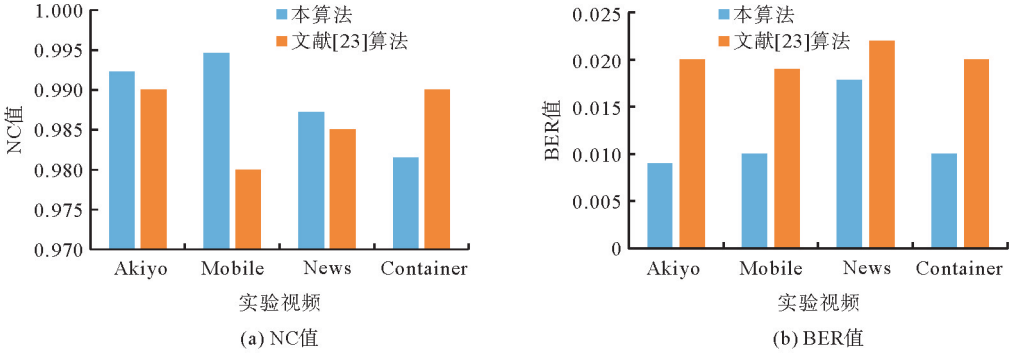


图 10 重压缩攻击下本算法与文献[23]算法的 NC 值和 BER 值对比

Fig. 10 Comparison of NC and BER values with our algorithm and the algorithm in [23] under recompression attacks

文献[24]在 QP 值为 28 时对嵌入水印的视频进行重压缩实验,并在 QP 值从 24 变化至 22 时进行重量化实验,实验结果如表 5 所示。可以看出,无论是在重压缩攻击还是重量化攻击下,本算法均表现出更强的鲁棒性。

表 5 重压缩攻击和重量化攻击下本算法与文献[24]算法的 BER 值对比

Table 5 Comparison of BER values between our algorithm and the algorithm in reference [24] under recompression attacks and requantization attacks

实验视频	重压缩攻击		重量化攻击	
	文献[24]的 BER 值	本算法的 BER 值	文献[24]的 BER 值	本算法的 BER 值
Mobile	0.03	0.018 7	0.14	0.052
News	0.03	0.017 8	0.14	0.047
Container	0.08	0.010 0	0.19	0.011

4 结论

本研究提出一种基于 H.264/AVC 域的半脆弱视频水印算法。该算法首先将视频 I 帧的宏块区域划分为构造视频特征矩阵、嵌入认证码和嵌入帧号三个功能区域;接着将 I 帧的帧号转为二进制序列,嵌入在对应的 I 帧中;然后基于视频特征序列与水印信息生成认证码,最后将认证码嵌入指定子块。在篡改检测阶段,通过提取各帧的帧号可初步检测丢帧、帧交换等攻击,进一步通过提取水印图像可实现帧内容篡改的检测与定位。对本算法进行了攻击实验,算法的归一化相关系数 NC 值均在 0.95 以上,能够有效抵抗重压缩和重量化攻击,并能准确识别出丢帧、帧交换和帧添加等篡改行为。需要说明的是,当前水印提取过程仍需依赖子块位置等辅助信息,是一种非盲水印。未来工作研究重点是水印的盲提取方法,以进一步提高算法的实用性与灵活性。

参考文献:

[1] 王翌妃,周杨铭,钱振兴等.鲁棒视频水印研究进展[J].中国图象图形学报,2022,27(1):27-42.
WANG Yifei,ZHOU Yangming,QIAN Zhenxing,et al. Review of robust video watermarking[J]. Journal of Image and Graphics,2022,27(1):27-42.

[2] LIBY J J,JAYA T. HSV model based data hiding in video for watermark applications [J]. Journal of Intelligent and Fuzzy Systems,2021,41(2):2731-2742.

[3] YASIN H M,SALLOW A B,MAHMOOD R Z. Efficient watermark embedding and extracting in raw digital video:Leveraging the least significant bit technique in the spatial domain[J]. International Journal of Intelligent Systems and Applications in Engineering,2024,12(1s):491-504.

[4] HAZIM H T,ALSEELAWI N,ALRIKABI H T H. A novel method of invisible video watermarking based on index mapping and hybrid DWT-DCT[J]. International Journal of Online & Biomedical Engineering,2023,19(4):155-173.

[5] FAN D,ZHANG X,KANG W S,et al. Video watermarking algorithm based on NSCT,pseudo 3D-DCT and NMF[J/OL]. Sensors,2022,22(13). DOI:10.3390/s22134752.

[6] FARRI E,AYUBI P. A robust digital video watermarking based on CT-SVD domain and chaotic DNA sequences for copyright protection[J]. Journal of Ambient Intelligence and Humanized Computing,2023,14(10):13113-13137.

[7] SUN W X,ZHAO H Y,ZHANG X,et al. Zero-watermarking algorithm for audio and video matching verification[J]. Aims Mathematics,2022,7(5):8390-8407.

[8] MISHRA A,BANSAL M,SHARMA A. Video watermarking of live streamed MPEG-4 frames using ELM-Fuzzy-PSO hybrid scheme[J]. Multimedia Tools and Applications,2023,83:41997-42035.

[9] FARFOURA M E,KHASHAN O A,OMAR H,et al. A fragile watermarking method for content-authentication of H.264-AVC video[J]. Journal of Internet Services and Information Security,2023,13(2):211-232.

[10] MOHAMMAD G,MOHAMMAD G. A low complexity system for multiple data embedding into H.264 coded video bitstream[J]. IEEE Transactions on Circuits and Systems for Video Technology,2020,30(11):4009-4019.

[11] KACZYNSKI M,PIOTROWSKI Z,PIETROW D. High-quality video watermarking based on deep neural networks for video with HEVC compression[J/OL]. Sensors,2022,22(19). DOI:10.3390/s22197552.

[12] KHMAG A. A robust watermarking technique for high-efficiency video coding (HEVC) based on blind extraction scheme [J/OL]. SN Computer Science,2021,2(4). DOI:10.1007/s42979-021-00729-y.

[13] RANA S. 3D video watermarking for MVD based view-synthesis and RST attack[J]. Multimedia Tools and Applications,2023,83:26775-26795.

[14] 易银城,冯桂.抗重压缩编码的 3D-HEVC 视频零水印算法[J].信号处理,2020,36(5):778-786.
YI Yincheng,FENG Gui. A video zero-watermark algorithm against recompression coding for 3D-HEVC[J]. Journal of Signal Processing,2020,36(5):778-786.

[15] NGUYEN T S. Reversible data hiding scheme based on coefficient pair mapping for videos H.264/AVC without distortion drift[J/OL]. Symmetry,2022,14(9). DOI:10.3390/sym14091768.

[16] FAN D,ZHAO H Y,ZHANG C Y,et al. Anti-recompression video watermarking algorithm based on H.264/AVC[J/

OL]. Mathematics,2023,11(13). DOI:10.3390/math11132913.

[17] HAMMAMI A,BEN HAMIDA A,AMAR C. Blind semi-fragile watermarking scheme for video authentication in video surveillance context[J]. Multimedia Tools and Applications,2020,80(5):7479-7513.

[18] ASIKUZZAMAN M D,MAREEN H,MOUSTAFA N,et al. Blind camcording-resistant video watermarking in the DTC-WT and SVD domain[J]. IEEE Access,2022,10:15681-15698.

[19] ZHANG W W,ZHAO C,HUANG D T,et al. Semi-fragile video watermarking algorithm for H.264/AVC based on cost strategy[J]. Journal on Communications,2015,36(10):110-118.

[20] 陶睿欣.基于压缩域的鲁棒视频水印算法研究[D].武汉:中南民族大学,2021.
TAO Xinrui. Research on robust video watermarking algorithm based on compressed domain[D]. Wuhan:South-Central Minzu University,2021.

[21] OGLA R,MAHMOOD E S,AHMED R I,et al. New fragile watermarking technique to identify inserted video objects using H.264 and color features[J]. Computers,Materials and Continua,2023,76(3):3077-3095.

[22] FAN D,SUN W X,ZHAO H Y,et al. Audio and video matching zero-watermarking algorithm based on NSCT[J/OL]. Complexity,2022,8. DOI:10.1155/2022/3445583.

[23] LI C,YANG Y,LIU K,et al. A semi-fragile video watermarking algorithm based on H.264/AVC[J/OL]. Wireless Communications and Mobile Computing,2020. DOI:10.1155/2020/8848553.

[24] TIAN L H,DAI H T,LI C. A semi-fragile video watermarking algorithm based on chromatic residual DCT[J]. Multimedia Tools and Applications,2020,79(3):1759-1779.

(责任编辑:傅　游)