

doi:10.11830/ISSN.1000-5013.201502004



采用 HEVC 的视频内容认证

张明辉, 冯桂

(华侨大学 信息科学与工程学院, 福建 厦门 361021)

摘要: 提出一种基于高效视频编码(HEVC)的视频内容认证算法,根据图像纹理特征产生特征码,将特征码用于修改帧间 8×8 编码单元的分割模式、帧间预测模式和运动向量,并保留最佳的编码单元分割模式及相应的预测模式和运动向量.实验结果表明:该算法对视频质量影响很小,嵌入水印后码率的变化也很小;同时,该算法具有较好的脆弱性,可以用于视频认证.

关键词: 高效视频编码;内容认证;运动向量;帧间预测模式

中图分类号: TP 391 **文献标志码:** A **文章编号:** 1000-5013(2017)05-0721-06

Video Content Authentication Scheme for HEVC

ZHANG Minghui, FENG Gui

(College of Information Science and Engineering, Huaqiao University, Xiamen 361021, China)

Abstract: In this paper, a video content authentication scheme based on high efficiency video coding (HEVC) has been proposed, the scheme used the feature codes generated from image texture to modify partition modes, inter-prediction modes and the value of motion vector of inter-frame 8×8 coding unit, and reserve the optimal coding unit splitting mode with corresponding prediction mode and motion vector. The experimental results show that our proposed algorithm has very small effect on video quality and bitrate. And our scheme can be used for authenticating video content owing to its good fragility.

Keywords: high efficiency video coding; content authentication; motion vector; inter-frame prediction mode

高效视频编码(high efficiency video coding, HEVC)是 ITU-T 视频编码专家组和 ISO/IEC 运动图像专家组联合推出的最新视频编码标准.在相同的感知视频质量下,HEVC 比现有的标准减少 50% 的比特率^[1-2].视频内容认证一般采用脆弱视频水印算法,与之相关的文献多以 H.264 为基础.王美华等^[3]提出修改帧内 4×4 块量化离散余弦变换(DCT)高频系数的奇偶性嵌入水印,然而奇偶性的脆弱性并不好. Xu 等^[4]选择中频部分嵌入水印,算法在嵌入过程中修改了非零系数,因此嵌入水印后码率增加较大. Zhang 等^[5]提出的可逆脆弱算法中,将水印嵌入 4×4 块最后一个非零量化 DCT 系数中,虽然算法是可逆的,但水印的嵌入区域不定,从低频到高频都有可能,因而脆弱性受限.除了帧内量化 DCT 系数外,帧间运动向量(motion vector, MV)也常用来实现脆弱水印嵌入.文献[6-7]选择修改 MV 的奇偶性嵌入水印. Li 等^[8]选择修改 P 帧或 B 帧满足条件的运动向量差的分量,实现水印嵌入,由于算法嵌入过程要满足一定条件,因而嵌入容量受限.以 HEVC 为平台的水印算法较少见.王家骥等^[9]提出了在纹理块区域,根据水印修改帧内预测模式的奇偶性,此方案能获得良好的视觉质量,只是奇偶性的脆弱性并不好.因此,本文提出一种基于 HEVC 的视频内容认证算法.

收稿日期: 2015-02-03

通信作者: 冯桂(1960-),女,教授,博士,主要从事信号与信息处理、通信系统信息安全理论、多媒体通信的研究. E-mail: fengg@hqu.edu.cn.

基金项目: 福建省自然科学基金资助项目(2014J01242, 2016J01306)

1 HEVC 编码特性

相比 H.264 编码标准中的宏块, HEVC 采用了一个新的编码结构——编码树单元(coding tree unit, CTU)作为基本处理单元. 根据图像的内容, 每个 CTU 被自适应地分割为编码单元(coding unit, CU). CU 的大小有 8×8 , 16×16 , 32×32 和 64×64 . HEVC 将每个 CU 划分成预测单元(prediction unit, PU), 如图 1 所示. 图 1 中, 帧间 8×8 CU 不允许采用 $N \times N$ 分割, 这是为了最小化最坏情况下的内存带宽; 而非对称分割, 只有在 $2N \geq 16$ 时才允许^[10]. 对于每个 PU, HEVC 采用了两种帧间预测模式, 分别为帧间模式和合并模式. 帧间模式采用高级运动向量预测, 此模式下要传输 MV 的值给解码端; 合并模式, HEVC 从候选列表中选出代价最小的合并候选, 此模式下只要传输合并索引给解码端, 并产生一个合并标志^[11].

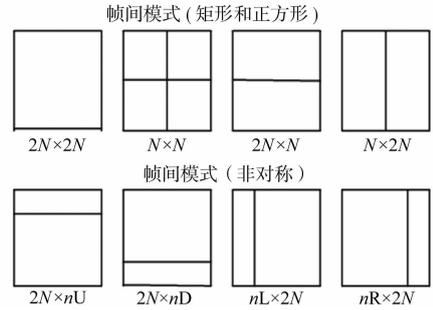


图 1 PB 分割模式

Fig. 1 Splitting mode for PB

2 算法的提出

2.1 特征码的产生

一般来说, HEVC 对纹理复杂的区域采用小尺寸的 CU; 对较平滑的区域采用大尺寸的 CU. 然而, 纹理特性在同一个场景中变化不大, 因此, 利用纹理特性产生特征码. S 表示区域平滑度, 定义为

$$S = 1 - 1/(1 + \sigma^2), \quad \sigma^2 = \mu_2, \quad (1)$$

$$\mu_n = \sum_{i=0}^{L-1} (z_i - m)^n p(z_i). \quad (2)$$

式(1), (2)中: z_i 为亮度的一个随机变量; $p(z)$ 为一个区域中的亮度级直方图; L 为亮度级数; $m = \sum_{i=0}^{L-1} z_i p(z_i)$ 是均值亮度; μ_n 为均值的 n 阶矩, 当 $n=2$ 时, μ_2 为区域中的亮度均方差, 用 σ^2 表示. 对于平滑区域, σ^2 较小, S 趋于 0; 对于纹理复杂区域, σ^2 较大, S 趋于 1. 将一帧图像划分为不重叠的 16×16 块, 计算每个块的平滑度 S , 定义一个阈值 T , 即

$$T = \alpha S_{\min} + (1 - \alpha) S_{\max}, \quad \alpha \in [0, 1]. \quad (3)$$

式(3)中: T 取值在 $[S_{\min}, S_{\max}]$ 之间, 根据 T 值划分出纹理块与非纹理块, $S > T$ 的块认为是纹理块. α 的取值要适中, α 过大时, T 趋于 S_{\min} , 会影响特征码的鲁棒性; α 过小时, T 趋于 S_{\max} , 影响纹理块的个数.

在编码过程中, HEVC 通常会将 16×16 的纹理块划分为 4 个 8×8 的 CU. 因而, 统计当前纹理块划分为 4 个 8×8 CU 的概率. 在 HM14.0 的平台上, 测试 4 个不同序列, 分辨率为 416×240 , 编码 30 个 I 帧, 每帧令 $\alpha=0.5$ 划分出纹理块. 在不同量化参数(quantization parameter, QP)下, 统计纹理块的划分情况, 如图 2 所示. 图 2 中: P 为概率.

由图 2 可知: 对于运动很缓慢的图像 BQsquare, 不同 QP 下纹理块被划分为 4 个 8×8 的概率在 98% 以上, 其他的测试序列也在 90% 以上, 说明纹理块划分具有很好的鲁棒性. 因而根据纹理块确定特征码, 即

$$m_{\text{code}} = \begin{cases} 1, & \text{纹理块,} \\ 0, & \text{非纹理块.} \end{cases} \quad (4)$$

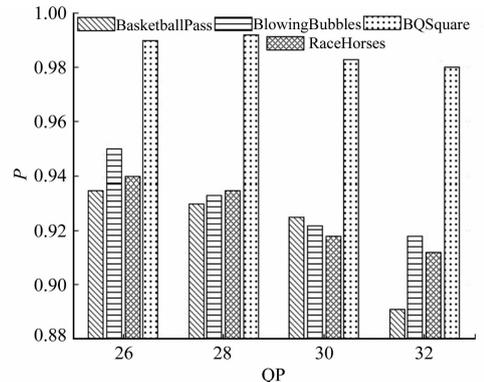


图 2 不同 QP 下纹理块划分为 4 个 8×8 CU 的概率

Fig. 2 Probability that texture block is divided into 4 CUs of 8×8 size under different QP

将特征码作为水印 $w = m_{\text{code}}$ 嵌入到 P 帧中.

2.2 水印嵌入

Liu 等^[12] 提出了一种用水印修改帧内和帧间分割模式的算法, 过程较为复杂. 文中将该算法简化以适应 HEVC 的分割模式, 并结合 HEVC 特有的帧间预测模式实现水印嵌入. 算法选择对 8×8 的 CU

进行修改, 具体有以下 5 个步骤.

步骤 1 在编码过程中, 如果当前 16×16 CU 的代价比 4 个 8×8 CU 的代价大, 则对这个 16×16 的 CU 进行重编码, 在重编码过程中嵌入水印. 这么做是为了减小算法对视频质量和码率的影响.

步骤 2 假设要嵌入的二值水印序列为 $W = \{\omega_1, \omega_2, \dots, \omega_k, \omega_{k+1}, \omega_{k+2}, \dots\}$, 如果当前水印 $\omega_k = 1$, 则在编码时跳过 $2N \times 2N$ 的分割模式; 反之, 则跳过 $2N \times N$ 和 $N \times 2N$ 分割模式.

步骤 3 对于修改后 CU 下的 PU, 如果当前水印值 $\omega_{k+1} = 0$, 则强制当前 PU 的预测模式为合并模式; 否则, 强制为帧间模式. 这一步处理只针第一个 PU, PU 序号, 如图 3 所示.

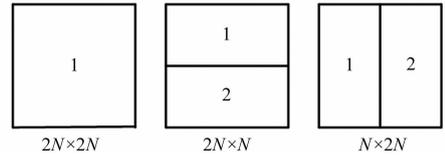


图 3 PU 的序号

Fig. 3 PU order

步骤 4 经过步骤 3 后, $N \times 2N$ 和 $2N \times N$ 的分割方式下 PU 会有剩余. 为了扩大容量, 对步骤 3 中剩下的 PU 也进行处理. 然而, 继续修改预测模式会对码率产生较大影响, 故这一步只处理剩下 PU 的 MV. Wang 等^[13]指出, 对 MV 进行加 1 或减 1 操作对视频质量影响很小. 因此, 如果当前 PU 的最佳预测模式为帧间, 就需要传输 MV 给解码端. 根据水印 ω_{k+2} 的值修改 MV 的水平分量 MV_x 和垂直分量 MV_y 中绝对值较大的分量的奇偶性, 令 $MV_{\max} = \max\{|MV_x|, |MV_y|\}$ 为水平和垂直分量中绝对值的最大值, 有

1) 如果 MV_{\max} 存在, 则

$$MV'_{\max} = \begin{cases} MV_{\max}, & \text{if } |MV_{\max}| \% 2 = 0 \text{ and } \omega_{k+2} = 0, \\ MV_{\max} + 1, & \text{if } |MV_{\max}| \% 2 = 1 \text{ and } \omega_{k+2} = 0 \text{ and } MV_{\max} > 0, \\ MV_{\max} - 1, & \text{if } |MV_{\max}| \% 2 = 1 \text{ and } \omega_{k+2} = 0 \text{ and } MV_{\max} < 0, \end{cases} \quad (5)$$

$$MV'_{\max} = \begin{cases} MV_{\max}, & \text{if } |MV_{\max}| \% 2 = 1 \text{ and } \omega_{k+2} = 1, \\ MV_{\max} + 1, & \text{if } |MV_{\max}| \% 2 = 0 \text{ and } \omega_{k+2} = 1 \text{ and } MV_{\max} > 0, \\ MV_{\max} - 1, & \text{if } |MV_{\max}| \% 2 = 0 \text{ and } \omega_{k+2} = 1 \text{ and } MV_{\max} < 0. \end{cases} \quad (6)$$

2) 如果 MV_{\max} 不存在, 则 $MV_x = MV_y$, 两者同时处理. 即

$$MV'_x = \begin{cases} MV_x, & \text{if } |MV_x| \% 2 = 0 \text{ and } \omega_{k+2} = 0, \\ MV_x + 1, & \text{if } |MV_x| \% 2 = 1 \text{ and } \omega_{k+2} = 0 \text{ and } MV_x > 0, \\ MV_x - 1, & \text{if } |MV_x| \% 2 = 1 \text{ and } \omega_{k+2} = 0 \text{ and } MV_x < 0, \end{cases} \quad (7)$$

$$MV'_x = \begin{cases} MV_x, & \text{if } |MV_x| \% 2 = 1 \text{ and } \omega_{k+2} = 1, \\ MV_x + 1, & \text{if } |MV_x| \% 2 = 0 \text{ and } \omega_{k+2} = 1 \text{ and } MV_x > 0, \\ MV_x - 1, & \text{if } |MV_x| \% 2 = 0 \text{ and } \omega_{k+2} = 1 \text{ and } MV_x < 0. \end{cases} \quad (8)$$

这里只列举 MV_x 的处理步骤, MV_y 的处理与其类似.

步骤 5 保留最佳的 CU 分割模式及其相应 PU 的预测模式和 MV 值.

2.3 水印的提取

水印提取不需要位置信息, 只需对帧间 8×8 CU 进行相应地提取, 是一种盲提取. 其具体提取分以下 3 个步骤.

步骤 1 如果当前 CU 的大小为 8×8 , 获得当前 CU 的分割模式, 分割模式为 $2N \times 2N$, 则提取水印 $m_k = 1$; 反之, 提取水印 $m_k = 0$.

步骤 2 由步骤 1 可知, 区分帧间预测模式的关键在于合并标志. 获取当前 PU 的合并标志, 如果合并标志位 true, 则提取水印 $m_k + 1 = 0$; 反之, 提取为 1.

步骤 3 在执行完步骤 2 后, 如果 CU 分割模式为 $N \times 2N$ 或 $2N \times N$, 则有 PU 剩余. 对于剩余的 PU, 如果其预测模式为帧间模式, 则继续从 PU 的 MV 进行提取, 即

1) 如果 MV'_{\max} 存在, 有

$$m_{k+2} = \begin{cases} 0, & \text{if } |MV'_{\max}| \% 2 = 0, \\ 1, & \text{if } |MV'_{\max}| \% 2 = 1. \end{cases} \quad (9)$$

2) 如果 MV'_{\max} 不存在, 则 $MV'_x = MV'_y$, 对其中任意一个分量进行提取, 即

$$m_{k+2} = \begin{cases} 0, & \text{if } |MV'_x| \% 2 = 0, \\ 1, & \text{if } |MV'_y| \% 2 = 1. \end{cases} \quad (10)$$

3 结果与分析

算法采用 HEVC 参考软件 HM14.0 进行测试. 原始 QP=30, 编码方式为 IPPP..., 总共编码 28 帧, 其余参数都为标准配置. 分别测试了 6 个具有 4 种分辨率的测试序列, 分别是 3 个 416×240 的序列 (BQSquare, BlowingBubbles, RaceHorses), 832×480 的序列 (BQMall), 1024×768 的序列 (ChinaSpeed) 和 1920×1080 的序列 (BasketballDrive).

3.1 视觉质量和不可见性分析

不同 QP(28,30)下, 有水印与无水印视频序列的亮度 R_{PSN} 值, 如表 1 所示. 定义 Diff 为原始视频序列 R_{PSN} 与嵌入水印后视频序列的 R'_{PSN} 的差值. Diff 值越小, 说明嵌入水印后的视频序列越接近原始视频序列. 由表 1 可知: 当 QP=28 时, Diff 的最大值发生在 BlowingBubbles 序列, 为 0.73 dB, 最小值发生在 BasketballDrive 序列, 为 0.01 dB; 当 QP=30 时, Diff 最大值为 0.07 dB, 最小值为 0.01 dB. 表明文中算法对视频质量的影响几乎可以忽略不计. 当 QP=30 时, 解码端原始 BQSquare 序列的第 10 帧图像及相应的含水印第 10 帧图像, 如图 4 所示. 通过对比图 4(a), (b) 可知: 人眼很难发现水印的嵌入, 说明该算法具有很好的不可见性. 当 QP=30 时, 原始序列与水印序列所有 28 帧的 R_{PSN} 值对比, 如图 5 所示. 图 5 中: N 表示帧数. 由图 5 可知: 每一帧的 R_{PSN} 差异都很小.

表 1 不同 QP 下的 R_{PSN} 值

Tab. 1 R_{PSN} values corresponding to different QP

dB

测试序列	QP=28			QP=30		
	R_{PSN}	R'_{PSN}	Diff	R_{PSN}	R'_{PSN}	Diff
BQSquare	33.85	33.83	0.02	32.58	32.52	0.06
BlowingBubbles	34.15	33.42	0.73	32.90	32.84	0.06
RaceHorses	34.54	34.51	0.03	33.08	33.06	0.02
BQMall	36.21	36.18	0.03	34.99	34.98	0.01
ChinaSpeed	38.81	38.79	0.02	37.36	37.29	0.07
BasketballDrive	38.15	38.14	0.01	37.49	37.48	0.01



(a) 原始序列的第 10 帧

(b) 含水印第 10 帧

图 4 原始序列和相应含水印的第 10 帧图像

Fig. 4 Original and watermarked 10st frame image

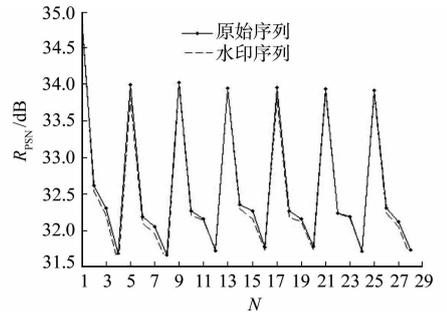


图 5 QP=30 下 28 帧 R_{PSN} 对比

Fig. 5 R_{PSN} values comparison of all the 28 frames for QP=30

3.2 比特率分析和容量测试

不同 QP(28,30)下, 无水印与有水印视频序列的比特率, 如表 2 所示. 表 2 中: R 和 R' 分别表示原始视频序列比特率和嵌入水印后视频序列的比特率; R_{var} 表示比特率的增长百分比, 定义为

$$R_{\text{var}} = \frac{R' - R}{R} \times 100\%. \quad (11)$$

由表 2 可知: R_{var} 的最大值发生在 QP=28 时的 BlowingBubbles 序列, 为 0.94%; R_{var} 的最小值发生在 QP=28 时的 BQMall 序列, 为 0.14%; 码率的增长不超过 1%, 说明文中算法对比特率的影响很小.

表 2 不同 QP 下的比特率
Tab. 2 Bitrate corresponding to different quantization parameters

测试序列	QP=28			QP=30		
	$R/\text{kbit} \cdot \text{s}^{-1}$	$R'/\text{kbit} \cdot \text{s}^{-1}$	$R_{\text{var}}/\%$	$R/\text{kbit} \cdot \text{s}^{-1}$	$R'/\text{kbit} \cdot \text{s}^{-1}$	$R_{\text{var}}/\%$
BQSquare	949.69	955.45	0.60	693.12	697.25	0.59
BlowingBubbles	771.22	778.47	0.94	566.37	570.44	0.71
RaceHorses	1 421.04	1 430.41	0.65	1 078.06	1 082.21	0.38
BQMall	2 135.81	2 139.01	0.14	1 627.27	1 631.43	0.25
ChinaSpeed	2 457.97	2 463.79	0.23	2 251.29	2 254.68	0.23
BasketballDrive	4 461.64	4 470.07	0.18	3 361.67	3 366.90	0.16
Average	2 031.39	2 039.53	0.46	1 596.30	1 600.49	0.38

不同 QP(28,30)下,28 帧的容量,如表 3 所示.由表 3 可知:对于每一个视频序列,QP 值越小,容量越大.这是由于 QP 越小,满足重编码条件的 16×16 块越多, 8×8 尺寸的 CU 个数越多,故容量越大.

3.3 视频认证

定义正确提取率 η ,于衡量算法的脆弱性,表示为

$$\eta = \frac{\text{right_bits}}{\text{total_bits}} \times 100\%. \quad (12)$$

式(12)中:right_bits 表示正确提取的比特数;total_bits 表示总的水印比特数. η 越小表明提取的水印与原始的水印差别越大,脆弱性越好.不同 QP 值含水视频重编码下的 η 值,如表 4 所示.表 4 中: η_{ave} 表示 η 的平均值.在重编码的过程中,P 帧 8×8 CU 的分割模式、帧间预测模式以及 MV 的值都有可能发生改变.对比不同的测试序列可知:ChinaSpeed 视频序列的 η_{ave} 值最小,为 0.43;RaceHorses 和 BQMall 视频序列的 η_{ave} 值最大,为 0.48.说明文中算法具有较好的脆弱性,能够用于视频认证.

表 4 不同 QP 下的 η 值
Tab. 4 η values under different QP

测试序列	无攻击	η						η_{ave}
		QP=24	QP=26	QP=28	QP=30	QP=32	QP=34	
BQSquare	1	0.48	0.47	0.43	0.51	0.48	0.47	0.47
BlowingBubbles	1	0.50	0.47	0.49	0.47	0.47	0.46	0.47
RaceHorses	1	0.52	0.53	0.52	0.53	0.50	0.47	0.48
BQMall	1	0.53	0.49	0.49	0.45	0.48	0.47	0.48
ChinaSpeed	1	0.45	0.43	0.42	0.42	0.45	0.44	0.43
BasketballDrive	1	0.45	0.46	0.43	0.45	0.47	0.46	0.45

3.4 对比测试

为了衡量文中算法的整体性能,将文中算法与文献[9]提出的算法作比较.采用与文献[9]相同的配置,周期为 16,GOP 为 8,编码 96 帧.对 BlowingBubbles 序列进行了测试,对比结果如表 5 所示.由表 5 可知:文中算法在 R_{PSN} 的变化上与文献[9]算法相差不多,码率的增长稍大;但文中算法的嵌入容量大很多,这是因为将水印嵌入了 P 帧,而文献[9]只将水印嵌入了 I 帧.同时,文中算法的脆弱性比文献[9]好.这主要是因为文献[9]只根据水印修改帧内预测模式的奇偶性,而文中算法综合了帧间分割模式、预测模式和 MV 的脆弱性.因此,文中算法的脆弱性更好.

表 5 不同算法性能对比结果
Tab. 5 Comparison results of performance of different algorithms

算法	Diff/dB	$R_{\text{var}}/\%$	C/bits	η
文献[9]算法	0.03	0.30	8 017	0.60
文中算法	0.02	1.02	43 428	0.48

4 结束语

针对重编码敏感,提出一种基于 HEVC 的脆弱水印算法.首先,根据图像的纹理特征产生特征码,将其作为水印嵌入 P 帧中;然后,根据水印值修改帧间 8×8 CU 的分割模式;最后,在不同的分割模式下,修改帧间预测模式和 MV 的值,保留最佳的 CU 分割模式及其相应的预测模式和 MV.实验结果表明:文中算法对视频质量的影响几乎可以忽略不计,嵌入水印后码率的增长也很小;同时,该算法具有较好的脆弱性,表明该算法能够用于视频内容认证.

参考文献:

- [1] SULLIVAN G J, OHM J, HAN W J, *et al.* Overview of the high efficiency video coding (HEVC) standard[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2012, 22(12): 1649-1668.
- [2] OHM J R, SULLIVAN G J, SCHWARZ H, *et al.* Comparison of the coding efficiency of video coding standards-including high efficiency video (HEVC)[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2012, 22(12): 1-12.
- [3] 王美华, 裴庆祺, 范科峰. 基于脆弱水印的 H. 264 视频完整性认证方案[J]. 西安电子科技大学学报, 2007, 34(5): 823-827.
- [4] XU Dawen, WANG Rangding, WANG Jicheng. A novel watermarking scheme for H. 264/AVC video authentication [J]. Signal Processing: Image Communication, 2011, 26(6): 267-279.
- [5] 张维伟, 张茹, 刘建毅, 等. 一种基于 H. 264/AVC 的视频可逆脆弱水印算法[J]. 电子与信息学报, 2013, 35(1): 106-132.
- [6] SAADI K S, BOURIDANE A, GESSOUM A. H. 264/AVC video authentication based video content[C]//5th International Symposium on Communications and Mobile Network. Rabat: IEEE Press, 2010: 1-4.
- [7] 杨高波, 李俊杰, 王小静, 等. 基于脆弱水印的 H. 264 视频流完整性认证[J]. 湖南大学学报(自然科学版), 2009, 36(6): 67-71.
- [8] LI Qian, WANG Rangding, XU Dawen. Fragile video watermarking algorithm based on H. 264/AVC[J]. Materials Research Innovations, 2013, 17(S1): 94-100.
- [9] 王家骥, 王让定, 李伟, 等. 一种基于帧内预测模式的 HEVC 视频信息隐藏算法[J]. 光电子·激光, 2014, 25(8): 1578-1585.
- [10] KIM I K, MIN J, LEE T, *et al.* Block partitioning structure in the HEVC standard[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2012, 22(12): 1697-1706.
- [11] LIN Jianliang, CHEN Yiwen, HUANG Yuwen, *et al.* Motion vector coding in the HEVC standard[J]. IEEE Journal of Selected Topics in Signal Processing, 2013, 7(6): 957-968.
- [12] LIU C H, CHEN O T. Data hiding in inter and intra prediction modes of H. 264/AVC[C]//International Symposium on Circuits and Systems. Seattle W A: IEEE Press, 2008: 3025-3028.
- [13] WANG Keren, ZHAO Hong, WANG Hongxia. Video steganalysis against motion vector based steganography by adding or subtracting one motion vector value[J]. IEEE Transactions on Information Forensics and Security, 2014, 9(5): 741-751.

(责任编辑: 黄晓楠 英文审校: 吴逢铁)