

# 基于运动向量修改的保持块间扰动最小的 HEVC 信息隐藏算法

唐小晶<sup>1,2</sup>, 马熠<sup>3</sup>, 余建昌<sup>1,2</sup>, 张弘<sup>1,2</sup>, 赵险峰<sup>1,2</sup>

<sup>1</sup> 中国科学院信息工程研究所 信息安全国家重点实验室 北京 中国 100195

<sup>2</sup> 中国科学院大学 网络空间安全学院 北京 中国 100195

<sup>3</sup> 北京信息技术研究所 北京 中国 100094

**摘要** 高效视频编码(High Efficiency Video Coding, HEVC)标准能够大幅度提升压缩效率, 正被广泛应用中。为提高基于运动向量修改的 HEVC 视频隐写算法的安全性, 本文结合了 HEVC 标准中帧间预测编码的新技术, 并利用在运动向量域进行信息嵌入扰动较小的优点, 提出了一种基于运动向量修改的保持块间扰动最小的 HEVC 信息隐藏算法。首先, 该算法量化了修改运动向量后对当前高级运动向量预测(Advanced Motion Vector Prediction, AMVP)单元和其相邻融合(Merge)模式预测单元产生的扰动代价, 将运动向量垂直分量和水平分量之和作为原始载体元素。其次, 结合校验网格码(Syndrome Trellis Codes, STC), 寻找扰动代价最小的嵌入路径, 通过修改运动向量中绝对值较大的分量将信息嵌入在 AMVP 模式的预测单元运动向量上。此外, 为减小对视频质量的影响, 利用修改后的运动向量更新运动向量差值。实验结果表明, 所提方法对视频序列进行信息嵌入后, 由于本文引入的扰动代价判断可以有效控制视频质量变化, 对视频序列的峰值信噪比、码率影响较小, 所以该方法对视频的扰动很小。而且本文算法不仅在视频质量保护上有良好的性能, 还对运动向量域专用隐写分析具有良好的抗检测能力, 有利于基于 HEVC 视频码流的隐蔽通信的进一步发展。

**关键词** 信息隐藏; 高效视频编码; 运动向量; 高级运动向量预测

中图分类号 TP391 DOI 号 10.19363/J.cnki.cn10-1380/tn.2022.12.07

## A HEVC Information Hiding Algorithm Based on Motion Vector Modification Minimizing the Distortion between Blocks

TANG Xiaojing<sup>1,2</sup>, MA Yi<sup>3</sup>, YU Jianchang<sup>1,2</sup>, ZHANG Hong<sup>1,2</sup>, ZHAO Xianfeng<sup>1,2</sup>

<sup>1</sup> State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100195, China

<sup>2</sup> School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100195, China

<sup>3</sup> Beijing Information Technology Institute, Beijing 100094, China

**Abstract** High Efficiency Video Coding (HEVC) standard can greatly improve compression efficiency, which is widely used. In order to improve the security of motion vector-based HEVC video steganography, the new technology of inter-frame prediction coding in HEVC is used in this paper. It also makes use of the advantage of less disturbance of information embedding in the motion vector domain, and then a motion vector-based HEVC information hiding algorithm to minimize the distortion between blocks is proposed in this paper. Firstly, the proposed algorithm defines the motion vector distortion of the current Advanced Motion Vector Prediction (AMVP) mode unit and its adjacent Merge mode units, and the sum of the vertical and horizontal components of the motion vector is taken as the original carrier element. Secondly, the Syndrome Trellis Codes (STC) is utilized to minimize the distortion. And then information hiding is realized by modifying components with large absolute values in the motion vector of selected AMVP mode units. In addition, to reduce the impact on video quality, the motion vector difference is also updated by using the modified motion vector. The experimental results show that: since the calculation method of disturbance in this algorithm can effectively reduce impact on video quality. The change of the peak signal-to-noise ratio and the bitrate is effectively controlled. The algorithm in this paper has little disturbance to the video. Moreover, the algorithm in this paper not only has good performance in video quality protection, but also performs well in resisting motion vector-based steganalysis, which is conducive to the further development of hidden communication based on HEVC video.

通讯作者: 马熠, 工程师, Email: mayi\_5501@126.com。

本课题得到国家重点研发计划(No. 2022QY0101)项目资助。

收稿日期: 2020-06-24; 修改日期: 2020-09-30; 定稿日期: 2022-12-07

**Key words** information hiding; high efficiency video coding; motion vector; advanced motion vector prediction

## 1 引言

信息隐藏指修改载体对象,使其包含需要传递的信息,形成载密对象。除接收者之外,其他人无法知道信息经过传递以及信息的内容<sup>[1]</sup>。现代信息隐藏所用载体包括数字图像、视音频、文本等多媒体文件。最近几年随着流媒体直播、网络视频服务的快速发展,数字视频因为其传播信息多、视觉表现丰富等优点,正逐步取代图像成为主流的传播媒介。因此,基于数字视频的信息隐藏算法的研究,正逐步成为信息隐藏领域的一大研究热点。目前流行的数字视频压缩编码标准有 AVS<sup>[2]</sup>、H.264/AVC<sup>[3]</sup>、H.265/HEVC<sup>[4]</sup>。HEVC 由 ISO/IEC MPEG 专家组和 ITU-T VCEG 专家组共同制定,又称 H.265 和 MPEG-H 第 2 部分。HEVC 达到 H.264/AVC 相同画面质量时,理论上最佳比特率只有后者的一半。

目前大多数视频信息隐藏算法在 H.264/AVC 压缩标准上实现<sup>[5-7]</sup>。相较于在 H.264/AVC 标准下实现信息隐藏,目前国内外基于 HEVC 标准下的信息隐藏算法的研究还不够深入。Wang 等<sup>[8]</sup>提出了一种基于帧内预测模式的隐写算法,通过修改帧内预测模式来进行信息嵌入,且根据帧内预测模式编码的性质,建立了三层隔离信道,避免了相邻的帧内预测单元在同一个信道内相互作用,抑制了嵌入变化对相邻信道的影响。但 HEVC 标准中帧内预测单元比较少,影响了该算法的嵌入容量。类似地,还有采用基于帧内预测模式调制的方法实现信息隐藏<sup>[9-10]</sup>。此外还有基于熵编码<sup>[11]</sup>和 DCT/DST<sup>[12-13]</sup>调制的信息隐藏方法。

目前在 HEVC 标准上,将信息嵌入到运动向量(Motion Vector, MV)的研究并不多。但由于预测残差会吸收运动向量修改产生的隐写扰动,所以在运动向量域进行信息嵌入,对视频质量和压缩效率等方面的影响会较小。所以有隐写算法结合 HEVC 中帧间预测的特点,对使用了高级向量预测(Advanced Motion Vector Prediction, AMVP)模式的预测单元(Prediction Unit, PU),通过修改其预测运动向量(Motion Vector Prediction, MVP)的索引值来实现信息嵌入<sup>[14]</sup>。然后根据修改后的 MVP 索引值来修改运动向量差值(Motion Vector Difference, MVD),保证了由 MVP 和 MVD 相加得到的 MV 不发生改变,从而可以避免影响视频质量。但在 HEVC 标准中的 AMVP 模式下,需要从 MVP 候选列表中选择编码比

特数较小的候选 MVP,将其作为当前 PU 的最终 MVP,再用 MV 减去最终 MVP 得到当前 PU 的 MVD。所以若通过直接修改 MVP 索引值来嵌入信息,会破坏选取编码比特数较小的 MVP 作为最终 MVP 这个特性,从而易被检测出视频经过篡改。此外,修改运动向量集合映射值来进行信息隐藏的方法<sup>[15]</sup>,实现了在  $N$  个运动向量分量中最多改变一个分量即可嵌入  $\lfloor \log_2(2N+1) \rfloor$  比特的效果,但由于严格限制了修改位置和修改操作,所以无法避免 MVD 出现(0, 0)的情况,这破坏了 AMVP 模式下 MVD 不能为(0, 0)的规则。

结合 HEVC 标准中帧间预测编码的新技术,以及在运动向量域进行信息嵌入扰动较小的优点,本文提出了一种保持块间扰动最小的信息隐藏算法。该算法将修改前后当前 PU 的率失真代价变化及相邻参考该 PU 的 Merge 模式 PU<sup>[16]</sup>的率失真代价变化之和作为当前 PU 的扰动代价,将运动向量垂直分量和水平分量之和作为原始载体元素,并结合校验网格码<sup>[17-18]</sup>(Syndrome Trellis Codes, STC)选出总扰动代价最小的嵌入路径。而后,通过修改其运动向量中绝对值较大的分量来嵌入秘密信息。最后,根据修改后的运动向量更新运动向量差值。

## 2 预备知识

### 2.1 HEVC 空间编码结构

为了达到更好的压缩效果,HEVC 标准采用了比 H.264/AVC 标准更加灵活的编码结构。在 H.264/AVC 等标准中,编码层的核心是宏块,一个宏块包含一个  $16 \times 16$  的亮度块,对于常用的 4:2:0 采样格式还包含两个  $8 \times 8$  的色度块。HEVC 标准则引入了编码树单元(Coding Tree Unit, CTU)、编码单元(Coding Unit, CU)、PU 以及变换单元(Transform Unit, TU)<sup>[19]</sup>。

HEVC 标准中的基本编码单位是 CTU,其大小范围为  $8 \times 8$  到  $64 \times 64$  像素。CTU 可以进一步划分为多个 CU, CU 的大小可为  $2N \times 2N$ , 其中,  $N$  可为 4、8、16 或 32。

而一个 CU 单元还可以进一步划分为 1 个、2 个或 4 个 PU, PU 则是进行帧内预测和帧间预测的基本单元。HEVC 标准为帧间预测的 CU 提供了 8 种划分 PU 的方式,分别为  $2N \times 2N$ 、 $N \times N$ 、 $2N \times N$ 、 $N \times 2N$ 、 $2N \times nU$ 、 $2N \times nD$ 、 $nL \times 2N$  和  $nR \times 2N$ , 如图 1 所示。

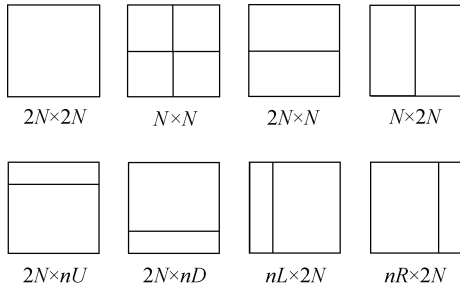


图1 帧间预测的 CU 划分 PU 的 8 种方式

Figure 1 Eight modes of CU with Inter-frame prediction divided into PUs

## 2.2 HEVC 运动向量预测模式

由于在空域或时域上, 相邻块之间的 MV 具有较强的相关性, 因此可利用相邻 PU 的 MV 对当前 PU 的 MV 进行预测。利用该思想, HEVC 中提出了 Merge 模式和 AMVP 模式<sup>[16]</sup>, 通过建立 MVP 候选列表, 从中选取满足特定条件的候选 MVP 作为当前 PU 对应的 MVP。

AMVP 模式下, 首先, 在 MVP 候选列表中选取率失真最小的 MVP 作为当前 PU 的 MV 搜索起点。其次, 在 MV 搜索起点附近搜索得到当前 PU 的 MV。最后, 选取 MVP 候选列表中编码比特数最小的候选 MVP, 作为当前 PU 对应的 MVP, 并用搜索得到的 MV 减去对应的 MVP, 得到当前 PU 的 MVD。而 Merge 模式下则是直接选取 MVP 候选列表中率失真代价最小的候选 MVP, 作为当前 PU 对应的 MVP, 并将其作为当前 PU 对应的 MV。由于在 Merge 模式下直接将 MVP 作为当前 PU 对应的 MV, 所以当前 PU 的 MVD 为(0, 0), 而 AMVP 模式下 MVD 不能为(0, 0), MVD 由当前 PU 搜索得到的 MV 减去 MVP 得到。在进行信息嵌入时, 若要对当前 PU 的 MV 进行修改, 其 MVD 也会受到影响。为使其 MVD 仍然等于当前 PU 的 MV 减去其 MVP, 本文算法在修改 MV 之后, 将利用修改后的 MV 减去当前 PU 的 MVP 得到更新后的 MVD, 从而实现信息嵌入。

### 2.2.1 AMVP 模式

AMVP 模式首先建立 MVP 候选列表。候选表的建立分为空域和时域两种情形。待空域和时域候选表建立后, 合并两者的候选 MVP, 保留列表中前两个候选 MVP 作为最终的候选列表, 先从中选择率失真代价最小的候选 MVP 作为运动向量搜索的起点确定当前 PU 的 MV, 再从中选择编码比特数较少的候选 MVP 作为最终的 MVP。下面将介绍具体流程。

#### 1) 候选列表的建立

在空域候选列表建立时, 假设当前 PU 的左下角

是  $A_0$ , 左侧是  $A_1$ , 左上角是  $B_2$ , 上方是  $B_1$ , 右上角是  $B_0$ , 如图 2 所示。AMVP 模式在当前 PU 的左侧和上方各产生一个候选 MVP, 组成空域候选列表。左侧选择候选 PU 的顺序是  $A_0 \rightarrow A_1$ , 上方选择顺序为  $B_0 \rightarrow B_1 \rightarrow B_2$ 。当左侧或上方检测到第一个可用 MV 时, 直接使用该 MV 作为当前 PU 的候选 MVP, 不再进行其他操作。当候选 MVP 对应的参考图像与当前 PU 不同时, 需要进行对应的比例伸缩, 计算公式如式(1), 其中,  $mv_{cur}$  表示当前 PU 的候选 MVP,  $d_c$  与  $d_s$  分别表示当前 PU 和候选 PU 到各自参考图像的距离,  $mv_{sel}$  为候选 PU 的 MV。

$$mv_{cur} = \frac{d_c}{d_s} mv_{sel} \quad (1)$$

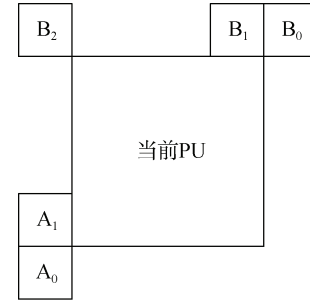


图2 候选 PU 相对位置

Figure 2 Relative position of candidate PUs

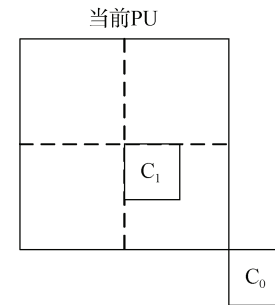


图3 同位 PU 位置

Figure 3 Position of co-located PUs

时域候选列表建立则利用了当前 PU 在邻近已编码图像中对应位置 PU(即同位 PU)的 MV, 同位 PU 的位置首选为  $C_0$ , 若  $C_0$  位置同位 PU 不可用, 则用  $C_1$  位置的同位 PU 进行替换, 如图 3 所示。由于同位图像的参考图像与当前 PU 不同, 所以也要进行比例调整, 此时式(1)中  $d_s$  对应同位 PU 所在图像到其参考图像的距离,  $mv_{sel}$  为同位 PU 的 MV。

#### 2) MVP 确定

待空域和时域候选列表建立后, 从空域候选列表选择 2 个候选 MVP, 从时域候选列表中选择 1 个候选 MVP, 然后合并相同的候选 MVP。若合并后候

选 MVP 少于 2 个, 则用(0, 0)填补, 若多于 2 个, 保留列表中的前两个候选 MVP, 作为最终的候选列表。最后从候选 MVP 列表中选择一个率失真代价最小的候选 MVP 作为运动向量搜索的起点确定 MV, 再从中选取编码比特数较少的候选 MVP 作为当前 PU 对应的最终 MVP, 率失真代价  $J$  计算如式(2)所示。

$$J = SAD(P_{cur}, P_{ref}) + \lambda R \quad (2)$$

其中,  $SAD$  为失真函数, 用于计算当前预测单元  $P_{cur}$  与其参考的预测单元  $P_{ref}$  之间的绝对误差和。  $R$  是 MV 相关信息(MV, 参考图像索引, MVP 索引等)的编码比特数,  $\lambda$  为拉格朗日因子, 用于控制失真和码率之间的平衡。

### 2.2.2 Merge 模式

Merge 模式的第一步也是建立 MVP 候选列表, 候选表的建立也分为空域和时域两种情形。待空域时域候选表建立后, 从空域和时域候选表中共挑选 5 个候选 MVP, 然后合并相同的候选 MVP, 最终从合并后的候选 MVP 列表里, 选择率失真代价最小的候选 MVP 作为当前 PU 的最终 MVP。

#### 1) 候选列表的建立

在时域候选列表的建立上, Merge 模式与 AMVP 模式有相同的规则。在空域候选列表的建立上, Merge 模式与 AMVP 模式的规则却有所不同, Merge 模式下的空域候选列表最多可以提供 4 个候选 MVP。

如图 2 所示, Merge 模式挑选候选块的顺序为  $A_1 \rightarrow B_1 \rightarrow B_0 \rightarrow A_0 \rightarrow (B_2)$ , 其中  $B_2$  为替补, 只有前面四个出现缺失时才会取  $B_2$ 。对于  $N \times 2N$ 、 $nL \times 2N$  或  $nR \times 2N$  的 CU 划分方式, 当前 PU 候选列表中不存在  $A_1$ , 这是由于若当前 PU 继续使用  $A_1$  的信息, 那么当前 PU 和其左边 PU 的 MV 将一致, 这将使  $N \times 2N$ 、 $nL \times 2N$  或  $nR \times 2N$  的 CU 划分方式与  $2N \times 2N$  划分方法无异, 从而使不同的 CU 划分方式失去意义。如图 4 所示, 对于  $N \times 2N$  的划分方式, 当前 PU 的候选列表中不会存在  $A_1$ 。

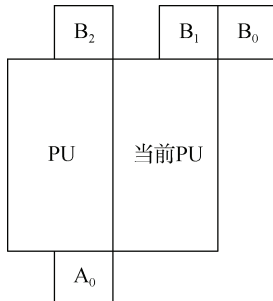


图 4 PU 的  $N \times 2N$  划分  
Figure 4  $N \times 2N$  division of PUs

同理, 对于  $2N \times N$ 、 $2N \times nU$  或  $2N \times nD$  的 CU 划分方式, PU 候选列表中不存在  $B_1$ 。如图 5 所示, 当前 PU 在选择候选 MVP 时将无法使用  $B_1$ 。

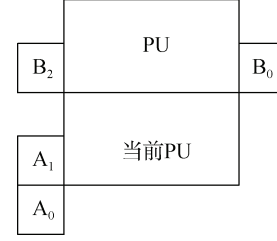


图 5 PU 的  $2N \times N$  划分  
Figure 5  $2N \times N$  division of PUs

#### 2) MVP 确定

空域和时域候选列表建立后, 从空域候选列表中最多选择 4 个候选 MVP, 从时域候选列表中最多选择 1 个候选 MVP, 然后合并相同的候选 MVP。若合并后候选 MVP 少于 5 个, 则用(0, 0)填补。

最终通过对比, 从合并后的 MVP 候选列表里选择率失真代价最小的候选 MVP, 作为当前 PU 的最佳 MVP。然后将得到的最佳 MVP 作为当前 PU 对应的 MV。

### 2.3 STC 编码

最小化总的嵌入失真可以提高隐写的安全性, 为最小化总嵌入失真, 我们将在嵌入信息的过程中采用 STC 编码<sup>[17]</sup>。本文隐写算法将取各个载体 PU 的 MV 水平和垂直分量之和的最低有效位作为原始载体元素并组成原始载体序列  $c$ ,  $c = (c_1, c_2, \dots, c_n) \in F_2^n$ 。 $m$  则代表秘密信息,  $m = (m_1, m_2, \dots, m_k) \in F_2^k$ ,  $n$  和  $k$  分别为原始载体序列和秘密信息的长度且  $n \geq k$ 。式(3)和式(4)分别描述了二元隐写编码的信息嵌入和提取映射。

嵌入映射

$$\text{Emb} : \{0, 1\}^k \rightarrow \{0, 1\}^n \quad (3)$$

提取映射

$$\text{Ext} : \{0, 1\}^n \rightarrow \{0, 1\}^k \quad (4)$$

将秘密信息  $m$  嵌入原始载体序列后, 将会得到隐密载体序列  $s$ ,  $s = (s_1, s_2, \dots, s_n) \in F_2^n$ 。用  $\rho_i$  代表原始载体元素  $c_i$  的扰动代价, STC 编码将尽可能使被修改载体元素的扰动代价和  $D(c, s)$  最小,  $D(c, s)$  定义如式(5)。

$$D(c, s) = \sum_{i=1}^n \rho_i |c_i - s_i| \quad (5)$$

嵌入和提取映射满足:  $\forall c \in \{0, 1\}^n, \forall m \in \{0, 1\}^k$ ,

$\text{Ext}_{\text{STC}}(\text{Emb}_{\text{STC}}(\mathbf{c}, \mathbf{m})) = \mathbf{m}$  成立。STC 编码的嵌入和提取的具体过程如下:

$$\text{Emb}_{\text{STC}}(\mathbf{c}, \mathbf{m}) = \arg \min_{\mathbf{s}' \in C(\mathbf{m})} D(\mathbf{c}, \mathbf{s}') = \mathbf{s} \quad (6)$$

$$\text{Ext}_{\text{STC}}(\mathbf{s}) = \mathbf{H}\mathbf{s} = \mathbf{m} \quad (7)$$

其中,  $C(\mathbf{m}) = \{z \in \{0, 1\}^n \mid \mathbf{H}z = \mathbf{m}\}$  是  $\mathbf{m}$  的陪集,  $\mathbf{H} \in \{0, 1\}^{k \times n}$  称为校验矩阵, 是收发双方共享的参数。

### 3 保持块间扰动最小的 MV 修改隐写算法

文中算法使用 HEVC 标准中的 AMVP 模式 PU 来进行信息隐藏。HEVC 标准中运动向量以水平分量和垂直分量进行存储, 文中算法结合 STC 编码以运动向量的水平和垂直分量之和作为信息的载体。如果运动向量水平分量绝对值远小于垂直分量绝对值, 则表示宏块的移动方向是偏向垂直方向的, 此时若运动向量水平分量发生微小变化将对视频质量产生严重影响<sup>[20]</sup>。所以当本文算法在 AMVP 模式 PU 上进行信息嵌入出现需要修改其 MV 的情况时, 则将修改其 MV 的水平或垂直分量中绝对值较大分量以减小影响。然后将修改前后该 PU 的率失真代价变化及相邻参考该 PU 的 Merge 模式 PU 的率失真代价变化之和作为当前 PU 的扰动代价, 利用 STC 找到一条扰动代价最小的嵌入路径。下面将具体介绍扰动代价的计算以及信息嵌入提取的基本流程。

#### 3.1 扰动代价计算

在扰动代价的计算上, 我们不仅考虑了修改前后该 AMVP 模式 PU 产生的率失真代价变化, 还考虑了相邻参考该 PU 的 Merge 模式 PU 的率失真代价变化。这是因为在嵌入秘密信息后, 某些 PU 的 MV 会被修改, 这将导致使用这些被修改 PU 作为预测参考的 Merge 模式 PU 受到影响, 其 MVP 会发生改变, 由于 Merge 模式下会直接将 MVP 作为当前 PU 对应的 MV, 因此 Merge 模式 PU 的对应 MV 则会发生改变。所以即使并没有直接将信息嵌入到使用 Merge 模式的 PU 的 MV 上, 某些 Merge 模式 PU 的 MV 还是因为其他 AMVP 模式 PU 的嵌入修改受到了影响。公式(8)为载体 PU 的扰动代价  $\rho_i$  的计算公式。

$$\rho_i = \min_{* \in \{+, -\}} \left\{ \left| J_i^A - \tilde{J}_{i*}^A \right| + \sum_{k=1}^N \left| J_k^M - \tilde{J}_{k*}^M \right| \right\} \quad (8)$$

其中,  $J_i^A$  表示当前载体 PU 在嵌入修改前的率失真代价,  $\tilde{J}_{i+}^A$  和  $\tilde{J}_{i-}^A$  分别表示对当前载体 PU 的 MV 中绝对值较大分量采取加 1 或减 1 操作修改后, 当前 PU 的率失真代价。  $J_k^M$  表示以当前载体 PU 作为预测参

考的 Merge 模式 PU 在嵌入修改前的率失真代价,  $\tilde{J}_{k+}^M$  和  $\tilde{J}_{k-}^M$  分别表示对当前载体 PU 的 MV 中绝对值较大分量采取加 1 或减 1 操作修改后, 以当前载体 PU 作为预测参考的 Merge 模式 PU 的率失真代价。  $N$  是以当前 PU 作为预测参考的 Merge 模式 PU 的个数。

#### 3.2 信息的嵌入

秘密信息的嵌入流程大致分为两个步骤, 首先, 遍历当前帧中使用 AMVP 模式的 PU, 将其作为载体 PU, 并对其 MV 进行预修改, 利用修改前后该 PU 和相邻参考该 PU 的 Merge 模式 PU 的率失真代价变化和, 分别计算对当前 PU 的 MV 中绝对值较大分量进行加 1 和减 1 操作的扰动代价。然后, 选取扰动代价较小的操作, 将其扰动代价作为当前 PU 对应的扰动代价, 并从当前 PU 的 MV 中提取水平和垂直分量之和的最低有效位作为原始载体元素。最后, 根据秘密信息  $\mathbf{m}$ 、原始载体序列  $\mathbf{c}$  和校验矩阵  $\mathbf{H}$  计算隐密载体序列  $\mathbf{s}$ , 将秘密信息嵌入到对应 PU 的原始载体元素上, 输出信息嵌入后的视频帧后, 继续下一帧处理。详细过程如算法 1 所示, 为简化描述,  $F$  代表当前原始的视频帧,  $\tilde{F}$  为信息嵌入的视频帧。步骤 1~5 为计算扰动代价, 6~9 为嵌入过程。

#### 算法 1 信息嵌入

输入:  $F$ 、 $\mathbf{m}$ 、 $\mathbf{H}$

输出:  $\tilde{F}$

1: 判断  $F$  是否为 I 帧,  $F$  不为 I 帧, 继续;  $F$  为 I 帧, 读取下一帧, 回到该步骤。

2: 按编码顺序选择  $F$  中 CU, 并依次判断 CU 划分的 PU 是否为 AMVP 模式, 是, 执行步骤 3; 否, 跳到步骤 4。

3: 读取当前 PU 的 MV, 对 MV 进行预修改, 利用公式(8)分别计算对当前 PU 的 MV 中绝对值较大分量进行加 1 和减 1 操作的扰动代价。然后选取扰动代价较小的操作, 并取其扰动代价作为当前 PU 的扰动代价。以 MV 的水平和垂直分量之和的最低有效位作为原始载体元素  $c_i$ 。

4: 判断当前 PU 是否为  $F$  中最后一个 PU, 是, 则执行步骤 5; 否, 回到步骤 2 判断下一个 PU。

5: 根据秘密信息  $\mathbf{m}$ , 原始载体序列  $\mathbf{c}$ , 以及收发双方共享的校验矩阵  $\mathbf{H}$ , 据式(5)计算得到各载体元素的扰动代价和, 在根据公式(6)选择使扰动代价和最小的隐密载体序列  $\mathbf{s}$ 。

6: 按编码顺序选择  $F$  中 CU, 并依次判断 CU



划分的 PU 是否为 AMVP 模式, 是, 执行步骤 7; 否, 跳到步骤 9。

7: 根据当前隐密载体元素  $s_i$  对当前 PU 提取的原始载体元素  $c_i$  进行对应修改。若  $s_i$  等于  $c_i$ , 则无需修改 MV。若  $s_i$  不等于  $c_i$ , 则对当前 PU 的 MV 中绝对值较大分量进行损失较小的加 1 或减 1 操作。然后用修改后的 MV 减去当前 PU 的 MVP, 得到新的 MVD, 赋给当前 PU。

8: 判断修改后的 MVD 的水平分量和垂直分量是否都为 0, 若都为 0, 则对 MV 水平和垂直分量中绝对值较大分量进行加 2 或减 2 操作, 再减去当前 PU 的 MVP 即得到修改后的 MVD。

9: 当前 PU 是否为 F 中最后一个 PU, 是, 输出 F, 读取下一帧, 回到步骤 1, 直至最后一帧; 否, 回到步骤 6 判断下一个 PU。

选择水平分量和垂直分量之和的最低有效比特位作为原始载体元素, 而不是直接选择 MV 中绝对值较大分量的最低有效位。是由于在修改 MV 中绝对值较大分量的过程中, 如果其原水平分量和垂直分量的绝对值相等, 此时若选择在垂直分量上进行信息的嵌入修改并对其进行了减 1 操作, 将会导致垂直分量绝对值小于水平分量绝对值, 而在信息提取的时候, 却会选择绝对值较大的水平分量进行信息提取, 这将导致秘密信息提取有误。

修改后, 要判断 MVD 是否等于 (0, 0), 由于, Merge 模式下 PU 的 MV 等于 MVP, 所以当 MVD 为 (0, 0) 时, 会使用 Merge 模式, 这会使当前 PU 与条件中选择的使用 AMVP 模式的 PU 作为嵌入载体相矛盾。故要再次对 MV 水平和垂直分量中绝对值较大分量进行加 2 或减 2 操作后, 使修改后的 MVD 不再等于 (0, 0)。要注意, 若是对 MV 中绝对值较大分量进行加 1 操作修改后 MVD 变为 (0, 0), 为使再次修改对 MV 扰动较小, 此时为使 MVD 不为 (0, 0), 则要选择对 MV 中绝对值较大分量进行减 2 的操作。同理, 若之前对 MV 进行了减 1 操作, 之后修正 MVD 时则选择对 MV 中绝对值较大分量进行加 2 的操作。

为使嵌入过程更加清晰, 下面给出所提算法在 HEVC 标准视频中进行秘密信息嵌入的例子。图 6 为示例中 PU 的划分图, 其中, 阴影块代表待嵌入信息的 AMVP 模式 PU, 黑色块代表以 AMVP 模式 PU 为预测参考的 Merge 模式 PU, 白色块代表其他不以 AMVP 模式 PU 为预测参考的 Merge 模式 PU。图中序号为 4、5 的 PU 的预测参考是序号为 3 的 Merge 模式 PU, 若修改 3 号 PU, 会对 4、5 号 PU 的 MV 产

生影响。待嵌入秘密信息  $\mathbf{m} = (1, 0, 1)^T$ , 校验矩阵为

$$\mathbf{H} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}。$$

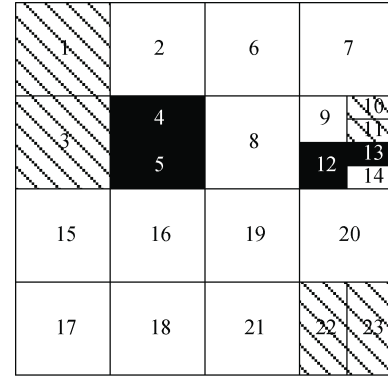


图 6 PU 划分

Figure 6 Division of PUs

从符合条件的 AMVP 模式 PU 中提取原始载体元素并组成原始载体序列。示例中, 1、3、10、11、22、23 号 PU 为 AMVP 模式 PU, 即载体 PU。如表 1 所示, 给出了进行信息嵌入之前载体 PU 的各个参数, 其中,  $\text{LSB}(\mathbf{c})$  代表原始载体序列, 其由 1、3、10、11、22、23 号 PU 的 MV 水平和垂直分量之和的最低有效位组成。由表 1 可知, 原始载体元素序列为  $(1, 1, 1, 0, 1, 1)^T$ 。

表 1 信息嵌入前 PU 的参数

PU	MV	MVP	MVD	LSB(c)
1	(-63, -4)	(-63, -2)	(0, -2)	1
3	(-65, -2)	(-62, -3)	(-3, 1)	1
10	(79, -60)	(86, -60)	(-7, 0)	1
11	(76, -60)	(79, -60)	(-3, 0)	0
22	(-31, 38)	(-30, 34)	(-1, 4)	1
23	(-33, -22)	(-28, -24)	(5, 2)	1

通过扰动代价大小判断嵌入修改时进行加 1 还是减 1 操作, 表 2 给出了分别对各载体 PU 的原始载体元素进行加 1 和减 1 操作所得到的扰动代价, 并通过计算得到使总扰动代价最小的隐密载体序列  $\mathbf{s}$ 。从表中可以看到, 3 号和 22 号 PU 的原始载体元素需要被修改, 由 1 变为 0。选取扰动代价较小的操作对 PU 的原始载体元素进行修改, 由表中数据可得, 对 3 号 PU 扰动代价较小的操作为加 1 操作, 对 22 号 PU 扰动代价较小的操作为减 1 操作。

由表 2 选择出扰动代价较小的加 1 或减 1 操作后, 在信息嵌入过程中, 用选择的对应操作对当前

表 2 加 1 或减 1 的代价

Table 2 Cost of adding or subtracting one

PU	LSB(c)	$\rho^+$	$\rho^-$	$s$
1	1	312	345	1
3	1	45	98	0
10	1	136	154	1
11	0	201	189	0
22	1	107	52	0
23	1	153	144	1

(注:  $\rho^+$ 代表修改操作为加 1 的扰动代价;  $\rho^-$ 代表修改操作为减 1 的扰动代价)

PU 的 MV 中绝对值较大分量进行修改。如表 3 所示, 可看到进行信息嵌入之后, 序号为 3 和 22 的 PU 的运动向量和运动向量差值被更新。其中, 对 3 号 PU 的 MV 水平分量进行了加 1 操作, 对 22 号 PU 的 MV 垂直分量进行了减 1 操作, 并对其 MVD 也进行了相应的更新操作。

表 3 嵌入修改后的 AMVP 模式的 PU 的参数

Table 3 Parameters of PUs with AMVP after embedding

PU	LSB(c)	$s$	MV	MVP	MVD
1	1	1	(-63, -4)	(-63, -2)	(0, -2)
3	1	0	(-64, -2)	(-62, -3)	(-2, 1)
10	1	1	(79, -60)	(86, -60)	(-7, 0)
11	0	0	(76, -60)	(79, -60)	(-3, 0)
22	1	0	(-31, 37)	(-30, 34)	(-1, 3)
23	1	1	(-33, -22)	(-28, -24)	(5, 2)

### 3.3 信息的提取

秘密信息的提取步骤与信息嵌入的步骤类似, 具体算法步骤如算法 2 所示, 输入为某一视频帧  $F$ , 以校验矩阵  $H$ , 输出嵌入的秘密信息  $m$ 。

#### 算法 2 信息提取

输入:  $F$ 、 $H$

输出:  $m$

1: 判断  $F$  是否为 I 帧,  $F$  不为 I 帧, 继续;  $F$  为 I 帧, 读取下一帧, 回到该步骤。

2: 按解码顺序选择  $F$  中 CU, 并依次判断 CU 划分的 PU 是否为 AMVP 模式, 是, 执行步骤 3; 否, 跳到步骤 4。

3: 读取当前 PU 的 MV, 并提取 MV 的水平垂直分量之和的最低有效位, 赋予隐密载体元素  $s_i$ 。

4: 当前 PU 是否为  $F$  中最后一个 PU, 是, 则执行步骤 5; 否, 回到步骤 2 判断下一个 PU。

5: 将隐密载体序列  $s$  乘以校验矩阵  $H$ , 根据公式(7)提取出秘密信息  $m$ 。读取下一帧, 回到步骤 1, 直至处理完所有帧。

下文所举信息提取例子与上文给出的信息嵌入例子相对应, 解码后的 AMVP 模式 PU 即为嵌入了秘密信息的隐密 PU, 图 7 代表了对于隐密 PU 的提取, 其中, 阴影块则代表隐密 PU。

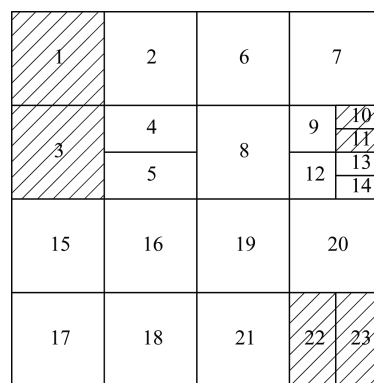


图 7 AMVP 模式的 PU

Figure 7 PUs with AMVP

表 4 解码后 AMVP 模式的 PU 的参数

Table 4 Parameters of PUs with AMVP after decoding

PU	MVP	MVD	MV	R	$s$
1	(-63, -2)	(0, -2)	(-63, -4)	-67	1
3	(-62, -3)	(-2, 1)	(-64, -2)	-66	0
10	(86, -60)	(-7, 0)	(79, -60)	19	1
11	(79, -60)	(-3, 0)	(76, -60)	16	0
22	(-30, 34)	(-1, 3)	(-31, 37)	6	0
23	(-28, -24)	(5, 2)	(-33, -22)	-55	1

根据解码后的 AMVP 模式隐密 PU 的 MV 得到隐密载体序列, 如表 4 所示, MV 列是由提取的隐密 PU 的 MVD 和 MVP 相加得到的运动向量。R 代表计算得到的运动向量水平和垂直分量之和, 取其最低有效位组成隐密载体序列  $s$ ,  $s$  为  $(1, 0, 1, 0, 0, 1)^T$ 。再将隐密载体序列  $s$  乘以校验矩阵  $H$ , 得到秘密信息  $m = (1, 0, 1)^T$ 。

## 4 实验结果与分析

本文算法在 HEVC 官方参考代码 HM16.16<sup>[21]</sup>进行了实现。实验中共采用了二十个不同的视频, 图 8 为原始视频帧序列图。每个视频编码 50 帧, 第一帧为 I 帧, 其余帧为 P 帧。实验中的所有嵌入信息均为



随机比特, 每帧的载体长度为 AMVP 模式 PU 的个数, 实验测试了在修改率分别为 0.25 和 0.5 时以及量化参数(Quantization Parameter, QP)分别为 27 和 36 时所提算法的性能, 修改率即被修改的 MV 与所有 MV 的比率。下面本文将从视频质量、码率变化、安全性这几个方面对嵌入前后的视频进行具体分析, 并对本文算法的嵌入效率和时间复杂度进行分析。

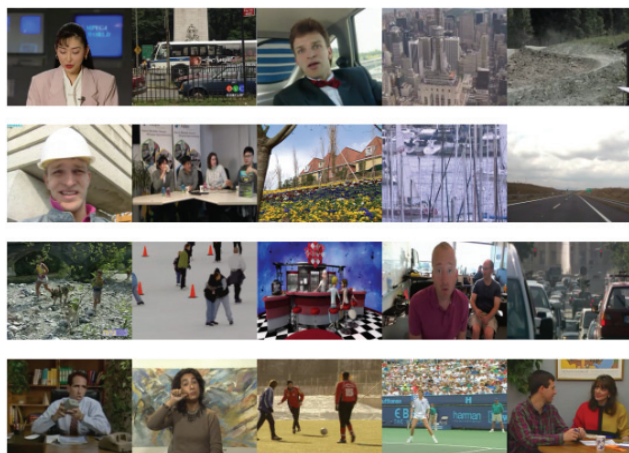


图 8 原始视频帧序列

Figure 8 Sequences of original video frames

#### 4.1 隐写的不可见性分析

本文将在主观上和客观上对隐写前后的视频进行分析。主观分析即通过观察隐写视频是否有肉眼可辨的失真来判断秘密信息是否存在, 客观分析即对隐写前后的视频质量扰动进行定量分析。

图 9 给出了实验中的九个视频嵌入前后某帧的对比图, 左侧为原始视频帧, 右侧为嵌入信息后的视频帧, 从上至下的视频帧依次取自 Akiyo、Dirtbike、Four\_people、Garden、Husky、Niklas、Salesman、Silent、Students。图中可以发现: 视频序列在信息嵌入后与原视频对比, 未出现能用肉眼察觉出的块错位、锯齿以及马赛克现象。隐写前后的视频在视觉效果上几乎没有差异, 所以本文所提隐写算法具有较好的视觉不可见性。

接着对视频进行客观分析, 常见的客观评价指标有峰值信噪比(Peak Signal to Noise Ratio, PSNR)、均方根误差(Mean Square Error, MSE)、信噪比(Signal to Noise Ratio, SNR)。本文将使用 PSNR 指标对信息嵌入前后的视频质量进行客观分析对比, 并且和 Li<sup>[15]</sup>提出的修改运动向量集合映射值进行信息隐藏的方法进行了对比。表 5 给出了在修改率为 0.25 和 0.5 的情况下, 未嵌入信息与嵌入信息的视频帧的平均 PSNR 值, 以及嵌入前后 PSNR 值的变化率。

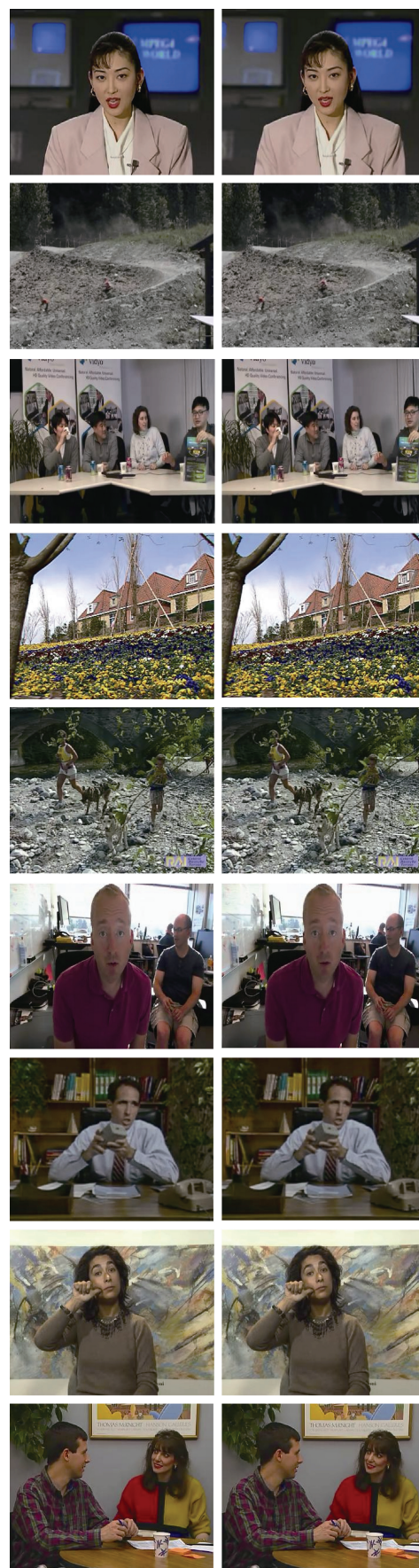


图 9 信息嵌入前后部分视频对照图

Figure 9 Video frames before and after embedding



表 5 信息嵌入前后部分视频的 PSNR 对比  
Table 5 Comparison of PSNR before and after embedding

视频序列	原始视频 PSNR(dB)	本文算法				Li 的算法			
		不同修改率的 PSNR(dB)		PSNR 变化率(%)		不同修改率的 PSNR(dB)		PSNR 变化率(%)	
		0.25	0.5	0.25	0.5	0.25	0.5	0.25	0.5
Akiyo	41.99	41.99	42.00	0.00	0.03	41.97	41.97	-0.05	-0.05
Dirtbike	38.01	38.01	38.02	0.00	0.03	38.00	38.01	-0.03	0.00
Four_people	39.66	39.65	39.65	-0.03	-0.03	39.65	39.64	-0.03	-0.05
Garden	36.85	36.85	36.85	0.00	0.00	36.85	36.83	0.00	-0.03
Husky	36.97	36.96	36.97	-0.03	0.00	36.96	36.95	-0.03	-0.03
Niklas	40.29	40.29	40.29	0.00	0.00	40.27	40.29	-0.05	0.00
Salesman	41.52	41.52	41.52	0.00	0.00	41.50	41.51	-0.05	-0.03
Silent	38.14	38.13	38.12	-0.03	-0.05	38.12	38.08	-0.05	-0.15
Students	39.01	39.01	39.01	0.00	0.00	39.01	39.01	0.00	0.00

从表 5 中可知, 本文算法对视频的 PSNR 值影响很小, 即使在 0.5 的修改率下, 视频序列的 PSNR 值最大波动也不超过 0.05%。而 Li<sup>[15]</sup>提出的算法对 PSNR 值的影响也较小。

4.2 不同修改率下的码率测试

对码率的测试分析, 也是检测视频是否经过篡改的一个重要方法。表 6 为在修改率为 0.25 和 0.5 的情况下, 本文算法和 Li<sup>[15]</sup>的算法对部分实验视频进行嵌入后的码率变化。

从表中可以发现, 本文算法对视频进行嵌入后, 视频的码率增长较少, 对同一视频来说, 修改率为 0.25 的码率增长幅度比修改率为 0.5 的码率增长幅度小。其中, 码率增长幅度最小值为视频在 0.25 修改率时的 2%, 码率增长幅度最大值为视频在 0.5 修改率时的 6.3%。修改率为 0.25 时, 大部分视频的码率增长控制在 2%到 3%, 修改率为 0.5 时, 大部分视频

的码率增长控制在 3%到 4%。

而 Li 提出的算法对码率的影响, 对比本文算法对码率的影响普遍更大。且随着修改率的增大, 视频的码率增长幅度变化更大。本文所提算法对视频进行嵌入后, 对视频的码率影响较小。

4.3 安全性分析

隐写算法的安全性主要体现在抗隐写分析的能力上。Li<sup>[15]</sup>的算法由于严格限制了修改位置和修改操作, 所以无法避免 MVD 出现(0, 0)的情况, 这破坏了 AMVP 模式下 MVD 不能为(0, 0)的规则, 所以利用此规则对 Li 算法进行针对性隐写分析时, 视频易被检测出经过篡改。

本文将通过对比本文隐写算法在修改率为 0.5 及 0.25 时的抗隐写分析性能, 来考察算法的安全性。本文采用的隐写分析方法<sup>[22]</sup>能够基于绝对误差和 (Sum of Absolute Difference, SAD)的运动向量局部最

表 6 信息嵌入前后部分视频的码率对比  
Table 6 Comparison of bitrate before and after embedding

视频序列	原始视频码率 (kbps)	本文算法				Li 的算法			
		不同修改率下码率(kbps)		码率变化率(%)		不同修改率下码率(kbps)		码率变化率(%)	
		0.25	0.5	0.25	0.5	0.25	0.5	0.25	0.5
Akiyo	102.3	105.8	108.7	3.3	6.3	108.0	111.7	5.6	9.2
Dirtbike	289.4	295.9	297.9	2.2	2.9	298.8	306.0	3.2	5.7
Four_people	147.5	152.0	153.1	3.1	3.9	154.8	157.2	4.9	6.6
Garden	2071.3	2122.5	2134.8	2.4	3.1	2132.0	2196.4	3.0	6.0
Husky	3603.4	3676.3	3698.8	2.0	2.5	3686.4	3770.7	2.3	4.6
Niklas	446.3	466.7	472.3	4.6	5.8	476.3	492.4	6.7	10.3
Salesman	150.5	153.7	155.2	2.2	3.1	154.2	163.0	2.5	8.3
Silent	262.5	269.8	273.1	2.8	4.1	279.8	285.9	6.6	8.9
Students	178.2	184.6	187.5	3.6	5.2	185.1	190.7	3.9	7.0

优判定准则, 来提取 18 维的隐写分析特征, 对于检测采用运动向量域嵌入的隐写视频, 可以有效地检测到其运动向量是否被嵌入秘密信息。

表 7 抗隐写分析检测率

Table 7 Anti-steganalysis detection rate		
QP	修改率	检测率
27	0.5	0.71
	0.25	0.69
36	0.5	0.66
	0.25	0.59

表 7 给出了视频的 QP 分别为 27 和 36 时, 隐写算法的隐写分析检测率, 从表中数据可以看出, 在隐写修改率和 QP 分别为 0.25 和 36 时, 隐写分析对

于本文隐写算法的检测率只有 0.59, 检测率比较低。所以从隐写分析的检测率来看, 本文隐写算法对于抵抗运动向量域专用隐写分析安全性能良好。

4.4 算法的嵌入效率和时间复杂度分析

由于本文算法结合了 STC 来寻找最小扰动修改路径, 所以本文算法的嵌入效率比较高。设嵌入的秘密信息长度为  $l$ , 嵌入后载体修改个数为  $k$ , 嵌入效率即  $l / k$ 。通过 STC 算法, 本文算法可在较小的修改率下, 实现较大的嵌入效率。

本文算法的计算复杂度主要来源于 STC 编码。表 8 为采用本文算法进行秘密信息嵌入, 编码和解码部分嵌入视频所需的时间与原视频所需的时间对比, 本文算法的秘密信息嵌入在编码端实现, 提取在解码端实现。

表 8 部分视频编解码时间

Table 8 Encoding and decoding time of some videos				
视频序列	原视频		嵌入视频	
	编码时间(sec)	解码时间(sec)	编码时间(sec)	解码时间(sec)
Akiyo	458.1	0.9	803.4	1.3
Garden	1087.3	4.0	1501	7.6
Niklas	763.4	2.2	1114.6	3.9
Silent	563.1	1.3	987.6	2.1

5 结论

本文提出了一种保持块间扰动最小的 HEVC 信息隐藏方法。在传统修改运动向量来嵌入信息的基础上, 增加了一种新的扰动代价判断。该算法将秘密信息嵌入在采用 AMVP 模式的 PU 上, 并将修改前后当前 PU 的率失真代价变化及相邻参考该 PU 的 Merge 模式 PU 的率失真代价变化之和, 作为当前 PU 的扰动代价。然后结合 STC 寻找最小扰动修改路径, 来修改当前 PU 的运动向量并更新其运动向量差值, 从而实现信息嵌入。实验表明, 本文引入的扰动代价判断可以有效控制视频质量变化, 所以本文算法对视频的扰动很小, PSNR 值和码率的变化不明显。而且本文算法不仅在视频质量保护上有良好的性能, 还对运动向量域专用隐写分析具有良好的抗检测能力, 有利于基于 HEVC 视频码流的隐蔽通信的进一步发展。

致 谢 在此向给本文给予指导的老师、提供帮助的同学和给本文评审的专家表示感谢。

参考文献

[1] Provos N, Honeyman P. Hide and Seek: An Introduction to Steg-

anography[J]. *IEEE Security & Privacy*, 2003, 1(3): 32-44.  
[2] Wiegand T, Sullivan G J. The H.264/AVC Video Coding Standard [Standards in a Nutshell][J]. *IEEE Signal Processing Magazine*, 2007, 24(2): 148-153.  
[3] Fan L, Ma S W, Wu F, et al. Overview of AVS video standard[C]. *2004 IEEE International Conference on Multimedia and Expo*, 2005: 423-426.  
[4] Sullivan G J, Ohm J R, 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.  
[5] Hu Y, Zhang C T, Su Y T, et al. Information hiding based on intra prediction modes for H.264/AVC[C]. *2007 IEEE International Conference on Multimedia and Expo*, 2007: 1231-1234.  
[6] Liao K, Lian S G, Guo Z C, et al. Efficient Information Hiding in H.264/AVC Video Coding[J]. *Telecommunication Systems*, 2012, 49(2): 261-269.  
[7] Yang G B, Li J J, He Y L, et al. An Information Hiding Algorithm Based on Intra-Prediction Modes and Matrix Coding for H.264/AVC Video Stream[J]. *AEU - International Journal of Electronics and Communications*, 2011, 65(4): 331-337.  
[8] Wang Y, Cao Y, Zhao X F, et al. Maintaining Rate-Distortion Optimization for IPM-Based Video Steganography by Constructing Isolated Channels in HEVC[C]. *The 6th ACM Workshop on Information Hiding and Multimedia Security*, 2018: 97-107.  
[9] Wang J J, Wang R D, Li W, et al. An Information Hiding Algorithm

- for HEVC Based on Intra Prediction[J]. *Journal of Optoelectronics Laser*, 2014, 25(8): 1578-1585.  
(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] Xu J, Wang R D, Huang M L, et al. A Data Hiding Algorithm for HEVC Based on the Differences of Intra Prediction Modes[J]. *Journal of Optoelectronics-laser*, 2015, 26(9): 1753-1760.  
(徐健, 王让定, 黄美玲, 等. 一种基于预测模式差值的 HEVC 信息隐藏算法[J]. *光电子·激光*, 2015, 26(9): 1753-1760.)
- [11] Jiang B, Yang G, Chen W. A cabac based hevc video steganography algorithm without bitrate increase[J]. *Journal of Computational Information Systems*, 2015, 11(6): 2121-2130.
- [12] Chang P C, Chung K L, Chen J J, et al. A DCT/DST-Based Error Propagation-Free Data Hiding Algorithm for HEVC Intra-Coded Frames[J]. *Journal of Visual Communication and Image Representation*, 2014, 25(2): 239-253.
- [13] Zhang X Y, Au O C, Pang C, et al. Additional sign bit hiding of transform coefficients in HEVC[C]. *2013 IEEE International Conference on Multimedia and Expo Workshops*, 2013: 1-4.
- [14] Hu Y J, Gong W B, Liu B B, et al. Large-Capacity Lossless HEVC Information Hiding Based on Index Parameter Modification[J]. *Journal of South China University of Technology (Natural Science Edition)*, 2018, 46(5): 1-8.  
(胡永健, 龚文斌, 刘贝贝, 等. 修改标志位的大容量无损 HEVC 信息隐藏方法[J]. *华南理工大学学报(自然科学版)*, 2018, 46(5): 1-8.)
- [15] Yang J, Li S B. An Efficient Information Hiding Method Based on Motion Vector Space Encoding for HEVC[J]. *Multimedia Tools and Applications*, 2018, 77(10): 11979-12001.
- [16] Sze, Vivienne, Budagavi M, et al. High efficiency video coding (HEVC)[M]. Springer: Integrated circuit and systems, algorithms and architectures, 2014: 113.
- [17] Filler T, Judas J, Fridrich J. Minimizing embedding impact in steganography using trellis-coded quantization[C]. *Proc SPIE 7541, Media Forensics and Security II*, 2010, 7541: 38-51.
- [18] Filler T, Judas J, Fridrich J. Minimizing Additive Distortion in Steganography Using Syndrome-Trellis Codes[J]. *IEEE Transactions on Information Forensics and Security*, 2011, 6(3): 920-935.
- [19] 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.
- [20] Hao-Bin, Zhao L Y, Zhong W D, et al. A novel steganography algorithm based on motion vector and matrix encoding[C]. *2011 IEEE 3rd International Conference on Communication Software and Networks*, 2011: 406-409.
- [21] JCT-VC Source: HM16.16. <https://hevc.hhi.fraunhofer.de/trac/hevc/browser/tags/HM-16.16>, May. 2019.
- [22] Wang K R, Zhao H, Wang H X. 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.



**唐小晶** 2018年在山东大学计算机科学与技术专业获得学士学位。现在中国科学院大学网络空间安全专业攻读硕士学位。研究领域为多媒体安全。研究兴趣包括: 多媒体安全。Email: tangxiaojing@iie.ac.cn



**马熠** 2006年获得了数字信号处理硕士学位。目前在北京信息技术研究所工作。Email: mayi\_5501@126.com



**余建昌** 2018年在四川大学通信工程专业获得学士学位。现在中国科学院大学网络空间安全专业攻读硕士学位, 研究方向为信息隐藏。Email: yujianchang@iie.ac.cn



**张弘** 2017年在中国科学院大学信息安全专业获得博士学位。现任中国科学院信息工程研究所助理研究员。研究领域为多媒体安全、视频编解码、机器学习。研究兴趣包括: 媒体安全与智能分析。Email: zhanghong@iie.ac.cn



**赵险峰** 2003年在上海交通大学计算机系统结构专业获得博士学位。现任中国科学院信息工程研究所研究员、博士生导师。研究领域为信息保密与内容安全防护, 包括: 信息隐藏、隐蔽通信及其检测, 内容伪造取证, 数字水印与数字版权保护, 内容安全标识及其管控, 多媒体特定内容与目标识别, 对抗情况下的机器学习, 隐私保护等。Email: zhaoxianfeng@iie.ac.cn