

# 融合像素串匹配的 AVS2 全色度屏幕与混合内容视频编码算法

赵利平<sup>1),2)</sup> 周开伦<sup>2)</sup> 郭靖<sup>2)</sup> 蔡文婷<sup>2)</sup> 林涛<sup>2)</sup> 朱蓉<sup>1)</sup>

<sup>1)</sup>(嘉兴学院数理与信息工程学院 嘉兴 314000)

<sup>2)</sup>(同济大学超大规模集成电路研究所 上海 200092)

**摘 要** 新一代主流云计算与信息处理模式及平台中的一个主要问题是客户端与云服务器端(或客户端)之间的数据传输瓶颈问题,屏幕与混合内容编码技术以带宽最省、安全性高等优势,成为解决该问题的核心技术之一.该文在我国第二代数字音视频编解码技术标准(AVS2)的屏幕与混合内容编码(SMCC)扩展版中,提出了一种融合像素串匹配的全色度屏幕与混合内容编码技术,该技术已经被 AVS2-SMCC 扩展版工作草案稿采纳.针对目前 AVS2 不支持全色度 4:4:4 格式的特点,作者设计与实现了一种融合 4:4:4 格式的像素串匹配编码算法和 4:2:0 格式的帧内、帧间预测编码算法的全色度屏幕混合内容编解码框架与算法.在此框架下,首先提出了一种 4:2:0 格式的帧内、帧间编码算法与 4:4:4 格式的串匹配编码算法的率失真代价计算方式;然后给出了一种改进的像素串匹配算法及其关键技术,包括最佳参考串确定算法、基于  $N$  元组的快速哈希表搜索算法、基于分段截断二进制码的长度参数编码方案和基于  $N$  元组的位移参数编码方案.实验结果表明,对于 AVS2-SMCC 通用测试序列中的移动的文字和图形类别,提出的算法与 AVS2-P2 基准算法集成了色度上、下采样后支持 YUV4:4:4 格式的编解码器相比,在编码复杂度增加较少的情况下,对于无损全帧内(AI)配置  $Y$  分量、 $U$  分量、 $V$  分量平均 BD-rate 降低率分别可达 56.1%,97.2%,99.2%;对于有损低延迟 P(LDP)配置  $Y$  分量、 $U$  分量、 $V$  分量平均 BD-rate 降低率分别可达 41.8%,70.9%,72.9%,能大幅提高屏幕与混合内容的编码效率.

**关键词** 视频编码;数字音视频编解码技术标准;屏幕混合内容;串匹配;哈希搜索;熵编码

中图分类号 TP391 DOI号 10.11897/SP.J.1016.2018.02482

## Pixel String Matching for Full-Chroma Screen and Mixed Content Coding in AVS2

ZHAO Li-Ping<sup>1),2)</sup> ZHOU Kai-Lun<sup>2)</sup> GUO Jing<sup>2)</sup> CAI Weng-Ting<sup>2)</sup> LIN Tao<sup>2)</sup> ZHU Rong<sup>1)</sup>

<sup>1)</sup>(College of Mathematics, Physics and Information Engineering, Jiaxing University, Jiaxing, Zhejiang 314000)

<sup>2)</sup>(Institute of Very Large Scale Integration, Tongji University, Shanghai 200092)

**Abstract** With the development of the new generation of cloud computing and information processing mode and platform, Screen and Mixed Content Coding (SMCC or SCC) becomes a key technology with the lowest bandwidth and high security to solve the data transmission bottleneck problem from the cloud to the client or the client to the client. The screen contents with their diversity and comprehensiveness show significant differences from the traditional video contents. Therefore, how to build efficient screen content coding technology has become a challenging issue in video coding area. In recent years, SCC has attracted increasing attention of researchers from both academia and industry. In response to the demand on SCC, ISO/IEC MPEG and ITU-T

收稿日期:2016-05-26;在线出版日期:2017-03-12. 本课题得到国家自然科学基金(61871289,61601200)、浙江省自然科学基金(LY15F020039,LY18F020021,LQ18F020006,LY19F020015)、上海市自然科学基金(18ZR1440600)资助.赵利平,女,1984年生,博士研究生,讲师,中国计算机学会(CCF)会员,主要研究方向为视频编码、屏幕图像编码. E-mail: zhaoliping\_jian@126.com.周开伦,男,1977年生,博士研究生,讲师,主要研究方向为视频编码、超大规模集成电路的架构、设计与验证.郭靖,男,1993年生,硕士研究生,主要研究方向为屏幕图像编码.蔡文婷,女,1990年生,硕士研究生,主要研究方向为屏幕图像编码.林涛(通信作者),男,1958年生,博士,长江学者特聘教授,博士生导师,主要研究领域为视频编码、屏幕图像编码. E-mail: lintao@ tongji.edu.cn.朱蓉,女,1973年生,博士,教授,主要研究领域为智能信息处理.

VCEG are jointly developing an SCC version of high efficiency video coding (HEVC) standard. Although HEVC SCC has made significant progress in the past few years, continuous improvement of performance is still indispensable and very challenging. Two of the major tools adopted in HEVC SCC draft up to now are Intra Block Copy (IBC) coding and palette coding (PLT). IBC is efficient to code repeated identical patterns of a few fixed sizes with rectangle or square shapes in a picture, but is not flexible enough to code repeated identical patterns of varying sizes from a few pixels to a few thousands of pixels with a variety of shapes. PLT can code repeated identical patterns inside a CU using two types of intra-CU pixel-index string matching, but it cannot exploit non-local repeated identical patterns. On the other hands, pixel string matching technique has a generic, flexible and uniform coding system architecture to exploit both local and non-local repeated identical patterns with a variety of sizes and/or shapes in a wide range of generic screen content. A pixel string matching technique is proposed in AVS2 (the second generation of the Audio Video Coding Standard) SMCC (AVS2-SMCC) extension to improve coding efficiency of screen and mixed content. The technique has been adopted into AVS2-SMCC extension working draft. In order to support the full-chroma screen and mixed content coding in AVS2, a 4:4:4 format pixel string matching algorithm is integrated into the tradition 4:2:0 format intra/inter prediction coding framework. In this framework, firstly, a Rate Distortion Cost (RDcost) calculation method for 4:2:0 format intra/inter prediction algorithm and 4:4:4 format pixel string matching algorithm is designed and implemented, then the key technologies of an improved pixel string matching algorithm including the best reference string selection scheme,  $N$ -tuple based fast hash table search scheme, segment based truncated binary coding method of length parameter and  $N$ -tuple based coding method of offset parameter are given. Compared with the AVS2-P2 main profile integrated with chroma down and up sampling to support coding of YUV 4:4:4 format, experimental results show that, for text and graphics with motion of AVS2-SMCC test sequences, the proposed algorithm achieves the average BD-rate reduction up to 56.1%, 97.2%, 99.2% in All Intra (AI) configuration and 41.8%, 70.9%, 72.9% in Low Delay P (LDP) configuration for Y, U, V component respectively at very low additional encoding complexity.

**Keywords** video coding; audio video coding standard; screen and mixed content; string matching; hash search; entropy coding

## 1 引言

新一代云计算与信息处理模式及平台中,直接传输云端产生的屏幕图像的像素数据到客户端是数据传输所需带宽最省、安全性高的一种系统架构<sup>[1]</sup>.如图 1 所示,客户端只需要一个显示屏和外设,屏幕在云端进行渲染,通过网络传输至客户端.只要传输的速度足够快,云端的计算资源与存储资源等就像在本地操作一样.在屏幕分辨率为  $1920 \times 1200$  像素、刷新率为 60 Hz 的情况下,图形处理单元显示屏输出的峰值数据传输率高达  $1920 \times 1200 \times 60 \times 24 = 3.1$  Gbps.因此,对屏幕图像应用超高压缩比和极高质量的数据压缩技术,即屏幕图像编码也称屏

幕内容编码(Screen Content Coding, SCC)技术<sup>[2]</sup>,成为当前云计算、云移动计算、桌面云、远程桌面、智能手机和平板电脑的第二显示、屏幕分享等新一代云计算与信息处理模式应用中解决客户端与云服务器端或客户端与客户端之间数据传输瓶颈问题中亟需的技术.

屏幕图像是指直接从各类计算机的屏幕图像显示单元捕获的视频或图像.作为一类包含自然图像在内的更普遍的视频图像,屏幕图像具有广泛的应用前景.屏幕图像的应用类别从视频编码质量需求的角度来分,至少包括以下两类:

(1) 高质量或超高质量的应用类别.以远程桌面、虚拟桌面框架、超瘦客户机、无线显示、仅有机顶盒功能的零客户机等为代表的云计算和云-移动计

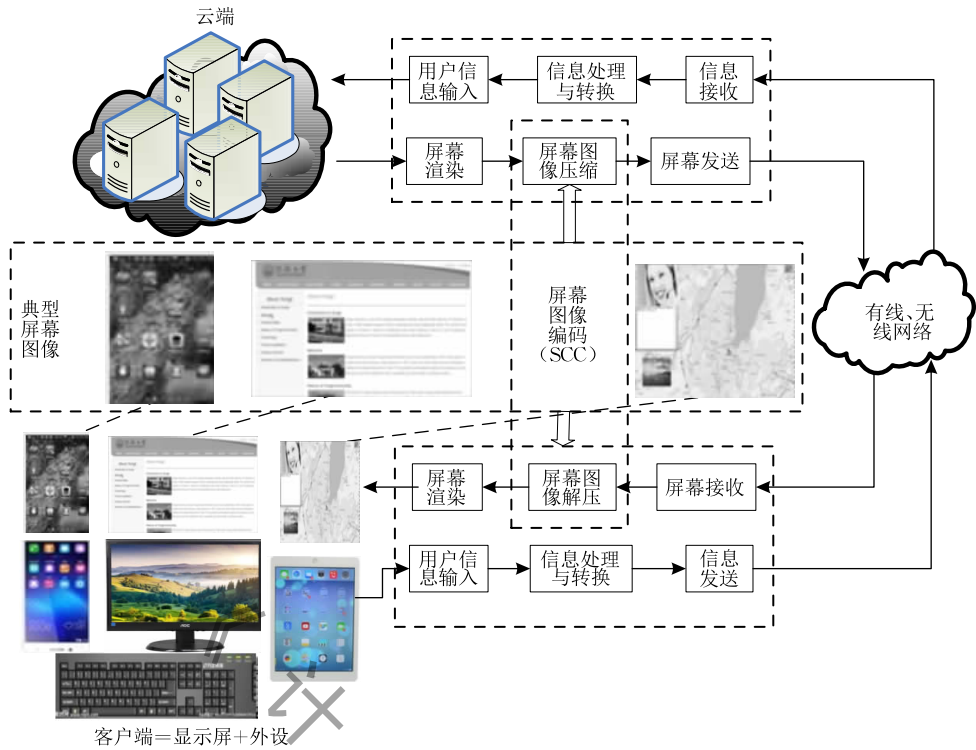


图 1 屏幕图像编码技术在新一代云计算与信息处理模式应用框图

算应用. 这类应用的一个特征是屏幕内容的使用者通常只有一个用户, 屏幕与用户之间的距离通常小于一米. 这类应用通常严格要求数学上无损或者视觉上无损的视频编码质量; 另一个特征是人机交互 (Human Computer Interaction, HCI) 的参与, HCI 过程需要对屏幕图像进行实时的编码与解码. 如图 1 所示, 客户端用户信息输入、云端信息的处理和屏幕渲染及其编码、压缩后的屏幕内容从云端传输至客户端、客户端屏幕内容的解码与显示一系列操作所需要的延迟应该在用户可以接受的范围内. 因此, 屏幕图像的编码时间与解码时间产生的延迟是整个系统响应时间中非常重要的因素, 对于 HCI 体验效果具有关键作用. 一般而言, 总的编码和解码延迟时间要求为 30 ms (小于每秒 30 帧的配置) 或更小. 因此, 这类应用中, 帧内编码性能比随机存取、帧间编码性能更加重要.

(2) 中等质量或低质量的应用类别. 以桌面共享、平板电脑和智能手机等移动装置的第二显示、高分辨率显示墙、远程教学、视频会议等为代表的桌面共享应用. 这类应用的特征是屏幕内容的使用者 (观看者) 通常是多个用户, 屏幕与用户之间的距离通常大于一米. 这类应用通常要求视觉上几乎无损, 在很多情形也能容忍一定程度上有损的视频编码质量.

与传统视频相比, 屏幕图像表现出显著不同

的特点. 从视频编码的角度来看, 屏幕图像最显著的特点是它的多样性和广泛性, 至少体现在以下 3 个方面:

(1) 一个编码块 (Coding Unit, CU) 中不同颜色的数目迥异, 可能出现介于下列两种极端情形之间的任何情形: ① 一个 CU 只有 1 种颜色; ② 一个 CU 中每个像素的颜色都不同, 即不同颜色的数目等于像素的数目.

(2) 图像中存在具有各种各样的形状、大小和相对位置的匹配图样.

(3) 一帧图像中匹配图样的数目很不相同, 可能具有任何数目的匹配图样, 有些图像中可以找到很多的匹配图样, 而有些图像中找不到任何匹配图样.

由于屏幕内容中也可能混合自然图像或视频, 屏幕内容也称为屏幕与混合内容 (Screen and Mixed Content Coding, SMCC). SCC 也称为 SMCC.

传统视频编码中的基于块匹配和变换的混合编码技术, 对非屏幕图像压缩效率比较高, 但对屏幕图像压缩效率不高, 必须寻求新的编码工具, 充分挖掘屏幕图像中的特征. 以大幅度提高屏幕图像编码效率.

因此, 高效的 SCC 算法受到了学术界和工业界的高度关注. 国际电信联盟 (ITU)、国际标准化组织

(ISO)和国际电工委员会(IEC)三大国际标准组织联合制定的 HEVC(High Efficiency Video Coding, HEVC<sup>[3]</sup>) SCC 扩展版<sup>[2,4-16]</sup>中重点研究的主要技术包括 IBC(Intra Block Copy)算法<sup>[4]</sup>、PLT(Palette)算法<sup>[5]</sup>、ILC(Intra Line Copy)算法<sup>[6]</sup>、串匹配(String Matching, SM)算法<sup>[7-16]</sup>等。

2015 年 8 月,我国数字音视频编码技术标准(Audio Video Coding Standard, AVS)正式启动了最新 AVS2 标准<sup>[17-18]</sup>的屏幕混合内容视频编码扩展版的制定工作<sup>[19]</sup>。

屏幕图像的多样性与广泛性使现有 HEVC SCC 扩展版中的 IBC 算法和 PLT 算法存在以下局限性:

(1)IBC 算法在一定程度上是将传统的帧间预测扩展到当前帧,是一种基于块匹配的算法,对于屏幕图像中重复出现的固定大小与形状的图样编码效率比较高,但是对于任意形状与大小的匹配图样缺乏灵活性。

(2)PLT 算法对 CU 内部的像素只能参考左边像素与上一行相同位置像素,且无法复制 CU 外部的同一帧屏幕图像中相同或相似的图样。

与 IBC 算法与 PLT 算法相比,SM 算法以“串”作为最小的匹配单元,其灵活的复制串大小(最小可以是一个像素或像素的一个分量)和形状与屏幕图像中具有各种大小和形状相似或者相同的图样具有天然的适应性,可以达到较高的匹配精度,从而提高屏幕图像的编码效率。以前的串匹配研究工作<sup>[7-16]</sup>主要在 HEVC 的框架中进行,围绕串匹配的最小匹配单元和匹配形状问题,串扫描方式问题,如何在参考范围内又快又好地找到更多匹配串问题,如何对串匹配参数进行优化熵编码问题而展开。

HEVC SCC 的编码性能已有很大提升,但同时编码复杂度也增加了很多倍,特别是远不能满足高清分辨率下软件实时编码的要求。而屏幕图像的应用场景大多都有实时编码的要求,本文在 AVS2-P2 的基础上,研究具有我国自主知识产权的屏幕混合内容的编码算法,提出了一种低复杂度、高效的融合像素串匹配的 AVS2 全色度屏幕与混合内容视频编码算法。此算法是 AVS2 提案<sup>[20]</sup>和提案<sup>[21]</sup>中的核心算法,已经被 AVS2-SMCC 工作稿草案采纳。

## 2 相关工作

串匹配算法的原型是混合色度采样率的双编码器(Dual-coder Mixed Chroma-sampling-rate, DMC)<sup>[7]</sup>,

此后成为了 HEVC SCC 的重点研究对象之一。

SM 算法按照不同的最小匹配单元,可以分为 3 类。第 1 类是分量串匹配算法<sup>[7-9]</sup>;第 2 类是索引串匹配算法<sup>[10]</sup>;第 3 类是像素串匹配算法<sup>[11-16]</sup>。SM 算法按照不同的匹配形状,可以分为 2 类。第 1 类是 1D SM 算法<sup>[7,9,11-12]</sup>,即匹配串与当前串具有相同的长度;第 2 类是仿 2D SM 算法<sup>[8,13-16]</sup>,即匹配串与当前串具有相同的长度且具有相同的二维形状。SM 算法根据 SM 模式的最小单位可以分为最大编码单元(Large Coding Unit, LCU)层次<sup>[7-8]</sup>和编码单元(Coding Unit, CU)层次的串匹配算法<sup>[11-16]</sup>;根据扫描的方向可以分为水平方式或垂直方式或水平、垂直自适应的扫描方式<sup>[9,13-16]</sup>;根据扫描的顺序可以分为光栅扫描顺序<sup>[7-9,11-12]</sup>与弓形扫描顺序<sup>[13-14,16]</sup>。

各种串匹配算法及其串匹配算法的改进算法主要围绕如何在参考范围内又快又好地找到更多匹配串问题,如何对串匹配参数进行优化熵编码问题而展开。在我们以前的研究<sup>[7-8,12-14,16]</sup>中,文献<sup>[12,16]</sup>分别提出了串匹配长度参数和位移矢量参数的优化熵编码算法。文献<sup>[13]</sup>提出了高效仿 2D 串匹配算法(Pseudo 2D String Matching, P2SM),在 CU 层次内融合了基于 PRB(Primary Reference Buffer)的 SM 算法和基于 SRB(Secondary Reference Buffer)的 SM 算法,采用哈希表搜索技术又快又好地找到匹配串,该算法与 HEVC SCC 算法相比,在编码时间增加很少的情况下,编码效率进一步提高。提案<sup>[14]</sup>提出了融合各种受限 SM 算法的通用串匹配算法(Universal String Matching, USM)。

以上串匹配算法均在 HEVC 的框架中探索与实现。本文在 AVS2-P2 基准档次(已进入国家标准的报批阶段,仅支持 YUV4:2:0 格式)框架下提出了融合像素串匹配的全色度屏幕混合内容编码算法,主要工作如下:

(1)针对目前 AVS2-P2 不支持全色度 4:4:4 格式的特点,设计与实现了一种融合 4:4:4 格式的像素串匹配编码和 4:2:0 格式的帧内、帧间预测编码的全色度屏幕混合内容编解码框架。在此框架下,提出了一种 4:2:0 格式的帧内、帧间编码的率失真代价值(Rate Distortion Cost, RDcost)计算方式与 4:4:4 格式的串匹配编码的 RDcost 计算方式;

(2)在 AVS2-P2 中实现了文献<sup>[13]</sup>中 PRB 的 SM 算法的基本技术;在此基础上,提出了一种改进的基于  $N$  元组(本文  $N$  取值为 8)的像素串匹配算法,包括最佳参考串确定算法、基于  $N$  元组的快速

哈希表搜索算法、基于分段截断二进制码的长度参数编码方案和基于  $N$  元组的位移参数编码方案。

### 3 融合像素串匹配的 AVS2 SCC 算法

#### 3.1 编解码框架

屏幕图像的原始采集数据一般为  $4:4:4$  格式(全色度格式). 另一方面, 目前 AVS2 基准档次中帧内、帧间模式支持的格式为  $4:2:0$  格式. 为了使  $4:2:0$  格式的编解码器能够处理  $4:4:4$  格式, 业界的一种主流方法是对  $4:4:4$  格式的原始图像进行色度下采样, 转换为  $4:2:0$  格式的图像后, 再用  $4:2:0$  格式的编解码器对其进行编码和解码, 对重构的  $4:2:0$  格式的图像进行色度上采样, 转换为  $4:4:4$  格式的重构图像.

本文采用这种方法使 AVS2 基准档次编解码

器也能够处理  $4:4:4$  格式的图像. 在进入 AVS2 基准档次编码器前, 将  $4:4:4$  数据经过色度下采样, 转换成  $4:2:0$  数据, 再进入编码器, 编码完成后, 再把重建的  $4:2:0$  的数据经过色度上采样转换为  $4:4:4$  格式的重构图像. 本文采用的色度下采样算法是  $4:2:0$  格式的色度分量取  $4:4:4$  格式中四个像素的色度分量的平均值(四舍五入), 而色度上采样算法则是重复色度分量四次.

融合像素串匹配算法的完整的 AVS2-P2 屏幕与混合内容视频编解码系统架构见图 2. 编码器端分为串匹配编码器子系统与基于块匹配的传统混合编码器子系统. 对于输入 CU  $O$ , 串匹配编码器进行编码形成位流  $b_1$  并获得重建 CU  $P_1$ , 与此同时, AVS2 基准档次中从传统帧内与帧间模式中选择出最优的模式, 对其进行编码形成位流  $b_2$  并获得重建 CU  $P_2$ . 两者经过 RDcost 比较, 最终选择其中的一种模式为最优的模式, 输出至位流中. 解码器原理类似.

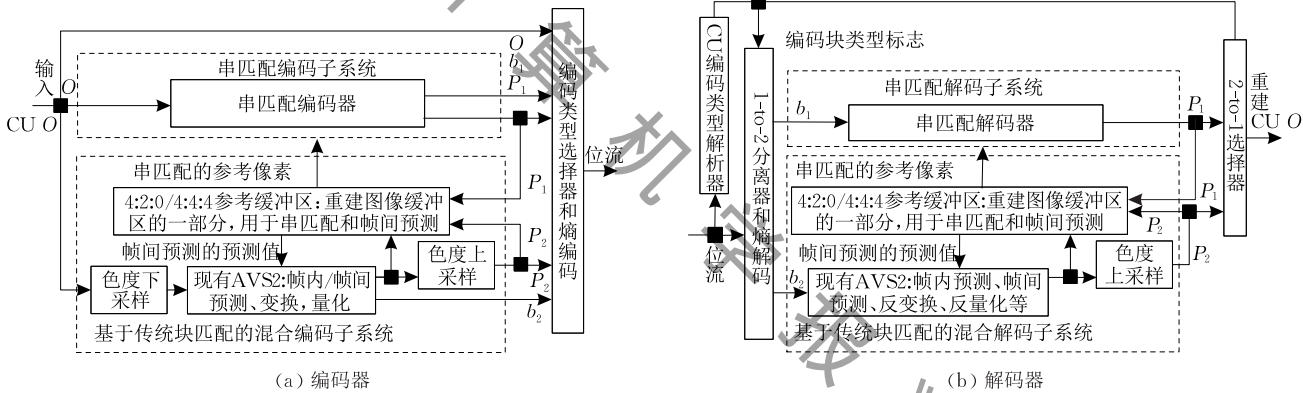


图 2 融合像素串匹配算法的 AVS2 全色度屏幕混合内容编解码系统框架图

在此框架下, 重构图像的失真包括编码失真和色度下采样失真. 对 AVS2 帧内与帧间编码器, 两种失真都存在; 对串匹配编码器, 仅存在编码失真. 为了解决该问题, 提出了一种计入这两种失真的  $4:2:0$  格式的帧内、帧间编码模式 RDcost 与  $4:4:4$  格式的串匹配编码模式 RDcost 的计算和比较方式.

支持  $4:4:4$  数据编码的基于  $4:2:0$  格式的 AVS2 帧内与帧间模式  $RDcost_{AVS2}$  计算方式见式(1).

$$RDcost_{AVS2} = (D_l + W_c \times D_c) + \lambda \times R_{AVS2} \quad (1)$$

其中  $D_l$  和  $D_c$  分别为亮度和色度失真度.  $W_c$  为帧内、帧间预测算法色度失真权重因子(本文取值为 0.2),  $R_{AVS2}$  为 AVS2 帧内或帧间模式编码的比特数,  $\lambda$  是 AVS2 中随量化因子变化的参数. 其中色度失真度  $D_c$  为原始的  $4:4:4$  的色度分量值与重建的  $4:4:4$

的色度分量值的差的平方和. 与传统的支持  $4:2:0$  格式数据编码的帧内、帧间预测算法 RDcost 计算方式不同的是:

(1) 色度失真度  $D_c$  不是直接从编码器中获得  $4:2:0$  的失真, 而是对其进行上采样后获得重建的  $4:4:4$  的色度分量值再与原始的  $4:4:4$  数据比较计算得到的失真.

(2) 引入了色度失真权重因子, 即  $W_c$ .

基于  $4:4:4$  的串匹配模式  $RDcost_{SM}$  计算方式如式(2)所示:

$$RDcost_{SM} = (D_l + w'_c \times D_c) + \lambda \times R_{SM} \quad (2)$$

其中  $w'_c$  为串匹配算法色度失真权重因子(本文取值为 0.3),  $R_{SM}$  为串匹配算法编码的比特数.

#### 3.2 像素串匹配算法的编解码基本操作

像素串匹配算法的基本操作是: 对当前 CU, 在

已重建的参考缓冲区内,以像素为单位,按照某种匹配顺序与规则,搜索最优参考串,最终将 CU 划分成一个或多个参考串和/或一个或多个未匹配像素.每一个参考串有两个匹配参数被写入位流:一个表示当前串与参考串之间的坐标偏移量的串位移矢量参数  $offset = (offsetX, offsetY)$  和一个串长度参数  $length$ . 如果从某个像素开始找不到任何匹配串,该像素就作为未匹配像素被写入位流.参考串与未匹配像素用一个匹配标志  $match\_flag$  来标识,其中  $match\_flag$  取值为 false 表示是一个未匹配像素,

否则表示是参考串.如图 3 所示, $CU_n$  按照水平弓形扫描的方式划分成三个参考串,分别是  $offset$  为 (9,4) 且  $length$  为 10 的参考串、 $offset$  为 (1,10) 且  $length$  为 32 的参考串和  $offset$  为 (9,0) 且  $length$  为 22 的参考串. $CU_{n+1}$  按照垂直弓形扫描的方式划分成一个未匹配像素 YUV(234, 234, 234) 和两个参考串,分别是  $offset$  为 (0,1) 且  $length$  为 7 的参考串、 $offset$  为 (-4,9) 且  $length$  为 56 的参考串.串像素匹配算法仅应用于大小为  $32 \times 32$ 、 $16 \times 16$  和  $8 \times 8$  的 CU.

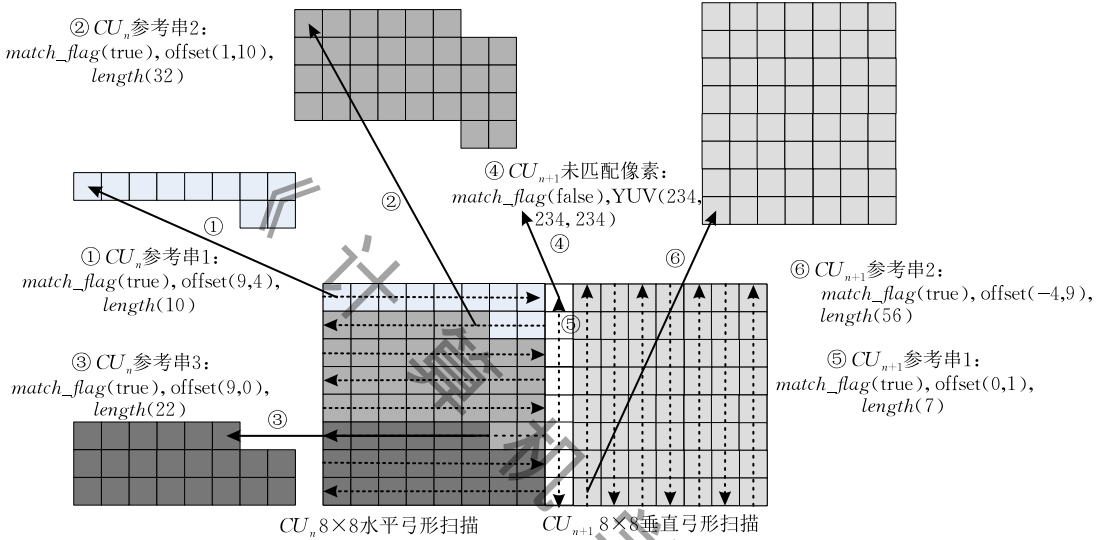


图 3 像素串匹配算法示意图

串匹配解码器的操作极其简单:首先解码 CU 的扫描方式,按照扫描顺序,逐个获得重建串或者重建像素直至整个 CU 重建完毕;解码  $match\_flag$ , 若当前  $match\_flag$  为 true,从码流中解码匹配参数  $length$  和  $offset$ ,将重建串的  $length$  个像素值从由  $offset$  确定的参考串位置复制到当前串位置;否则从码流中解码一个像素的值,该像素的值即重建像素.

### 3.3 像素串匹配类型码流的语法元素

像素串匹配算法是在 I 图像中加入了串匹配类型的编码单元,通过添加串匹配类型标识来表示是否是串匹配类型的编码单元.对于串匹配类型的编码单元,用  $cu\_sm\_coding()$  来定义语法元素,否则保持原 AVS2 中非串匹配类型的编码单元的语法元素定义.

表 1 给出了像素串匹配的基础语法元素.其中  $SizeInBit$  表示 CU 大小需要的比特数,  $PositionInPic$  表示宏块所在图像的位置.语法元素说明如下:

(1)  $scan\_mode\_flag$ :串匹配扫描模式标志.值

为 1 表示当前编码单元采用垂直扫描模式;值为 0 表示当前编码单元采用水平扫描模式.

(2)  $match\_flag$ :串匹配的匹配类型标志.值为 1 表示一个匹配串;值为 0 表示一个未匹配像素.

表 1 像素串匹配的基础语法元素

$cu\_sm\_coding()$ 的定义	描述符
$cu\_sm\_coding(SizeInBit, PositionInPic) \{$	
$numCodedPixel=0$	
$numPixelInCu=(1 \ll SizeInBit) \times (1 \ll SizeInBit)$	
$scan\_mode\_flag$	ae(v)
while ( $numCodedPixel < numPixelInCu$ ) {	
$match\_flag$	ae(v)
if( $match\_flag == 1$ ) { // a string	
$length\_minus1$	ae(v)
$offsetX$	
$offsetY$	
$numCodedPixel += (length\_minus1 + 1)$	
}	
else { // an unmatched pixel	
for( $cIdx=0; cIdx < 3; cIdx++$ )	
$unmatched\_pixel\_val$	u(8)
$numCodedPixel += 1$	
}	
} //while loop	
}	

(3)  $length\_minus1$ : 匹配串长度.  $length\_minus1$  的值加 1 表示当前匹配串的长度.

(4)  $offsetX$ : 匹配串的水平偏移.

(5)  $offsetY$ : 匹配串的垂直偏移.

(6)  $unmatched\_pixel\_val$ : 未匹配像素的分量的值. 8 位无符号整数, 表示当前未匹配像素的一个分量的值.

表 1 中  $ae(v)$  表示高级熵编码的语法元素.  $u(8)$  表示一个任意取值的字节. 语法元素  $length\_minus1$ 、 $offsetX$  和  $offsetY$  在熵编码的实现过程中, 可以根据其相关性与统计特性进行熵编码优化.

## 4 改进的像素串匹配算法关键技术

为了进一步提高屏幕图像的编码效率, 在文献 [13] 的 PRB SM 算法的基础上, 提出了一种改进的像素串匹配算法. 下面介绍其关键技术.

### 4.1 最佳参考串确定方案

在有损编码配置中, 当前串与参考串采用有损匹配, 即当前串的像素值与参考串的像素值允许一定的误差. 本文中的有损匹配误差设置值  $ErrorLimit$  是根据不同的量化参数 (Quantization Parameter, QP) 值而设定的, 计算方式如下:

$$ErrorLimit = floor((2^{16}/qScale)/2.0 + 0.5) + 1 \quad (3)$$

其中  $qScale$  是 AVS2 中的量化参数对应的量化权重因子.  $floor$  函数表示向下取整. 根据式 (3), QP 为 22、27、32、37 时,  $ErrorLimit$  取值分别为 8、11、17、26.

**定义 1.** 有损匹配准则  $C$ .

$$C = \begin{cases} \text{true,} & \text{参考像素与当前像素每个分量的} \\ & \text{差的绝对值小于等于 } ErrorLimit \\ \text{false,} & \text{否则} \end{cases} \quad (4)$$

**定义 2.** 参考串与当前串的相似度参数  $dist$ .

参考像素串与当前串的相似度参数  $dist$  定义为参考像素串与当前串所有像素三个分量的差值的平方和 (Sum of Squared Error, SSE).

$$dist = \sum_{c=1}^3 \sum_{j=1}^{length} ((refStr)^{c,j} - (curStr)^{c,j})^2 \quad (5)$$

其中  $(refStr)^{c,j}$  和  $(curStr)^{c,j}$  分别表示参考串和当前串的第  $c$  个分量第  $j$  个像素值.

设当前串  $curStr$  在参考范围内有  $n$  个起始匹配像素位置, 依次对每个位置进行像素串匹配, 可以获得  $n$  个参考串 (记为  $refStr_i$ ) 的匹配参数, 分别记为  $offsetX_i, offsetY_i, length_i, dist_i$ . 其中  $i$  的取值

范围为  $[1, n]$ .

**定义 3.** 参考串与当前串匹配度函数  $F$ .

参考串  $refStr_i$  与当前串匹配度函数  $F$  是根据参考串与当前串获得的一个或多个匹配参数、相似度参数的函数, 即

$$F(curStr, refStr_i) = f(offsetX_i, offsetY_i, length_i, dist_i) \quad (6)$$

**定义 4.** 最佳参考串评价规则  $R$ .

设  $p_1$  和  $p_2$  是两个起始匹配像素位置, 在  $p_1$  和  $p_2$  处进行像素串匹配, 分别获得  $refStr_1$  和  $refStr_2$ . 最佳参考像素串评价规则是根据两个参考串与当前串的匹配度函数  $F$  所获得的值判断哪个参考串作为当前串的匹配串. 即

$$R = \begin{cases} refStr_1, & \text{如果 } F(curStr, refStr_1) \text{ 大于} \\ & \text{等于 } F(curStr, refStr_2) \\ refStr_2, & \text{否则} \end{cases} \quad (7)$$

文献 [11] 中参考串与当前串的匹配度函数仅仅考虑了长度参数, 记为  $F'(curStr, refStr_i)$ , 见式 (8), 因此最佳参考串评价规则  $R$  是选择最长的串, 记为  $R'$ , 见式 (9).

$$F'(curStr, refStr_i) = Maximum(length_i) \quad (8)$$

$$R' = \begin{cases} refStr_1, & \text{如果 } length_1 \text{ 大于等于 } length_2 \\ refStr_2, & \text{否则} \end{cases} \quad (9)$$

该方案没有考虑匹配参数  $offset$  的影响 (有的参考串长度长一些, 但是参考位置却很远, 编码匹配参数消耗的比特数很多) 和有损编码中相似度参数. 为了解决该问题, 本文提出了参考串长度优先, 平均 RDcost 估计值最小的最佳参考像素串确定方案. 即提出了参考串与当前串匹配度函数  $F^{opt}$  和最佳参考串评价规则  $R^{opt}$ .

**定义 5.** 参考串与当前串平均 RDcost 估计值  $avg\_pRDcost$ .

参考串  $refStr_i$  与当前串 RDcost 估计值  $avg\_pRDcost_i$  是参考串与当前串中每个像素获得的匹配参数需要编码的比特数估计值与相似度参数值的加权组合, 见式 (10).

$$avg\_pRDcost_i = (\lambda \times bits(offsetX_i, offsetY_i, length_i) + dist_i) / length_i \quad (10)$$

其中  $bits(offsetX_i, offsetY_i, length_i)$  表示各匹配参数熵编码需要消耗的比特数, 此处的熵编码不考虑概率模型的影响, 因此是比特数的估计值.  $\lambda$  的取值同 AVS2 的 RDcost 的调节因子.

$$F^{opt}(curStr, refStr_i) = Maximum(length_i) \parallel Minimum(avg\_pRDcost_i) \quad (11)$$

$$R^{opt} = \begin{cases} refStr_1, & \text{if } (length_1 > aolength_2 \parallel \\ & (length_1 \geq length_2 - 1 \& \& \\ & avg\_pRDcost_1 < avg\_pRDcost_2)) \\ refStr_2, & \text{否则} \end{cases} \quad (12)$$

根据式(11)和(12),提出的最佳参考串评价规则  $R^{opt}$  在保留了选择最长的参考串作为最佳参考串的前提下,充分考虑了等长串或者次长串在匹配精度很高或者位置很近的参考串作为最终匹配串的可能性. AI 配置实验结果表明,对于 AVS2 SCC 通用测试序列<sup>[22]</sup>,提出的  $R^{opt}$  与  $R'$  相比, Y 分量 BD rate(Bjontegaard Delta rate<sup>[23-24]</sup>) 平均降低 4.5%.

#### 4.2 快速的哈希表搜索方案

像素串匹配算法的性能很大程度上取决于搜索的范围. 搜索的范围越大,找到的匹配串越多越好,编码效率也越高,同时编码的时间也越长. 基于哈希表的搜索技术<sup>[7-8,11,13]</sup> 是一种有效的解决方案,被用于大搜索范围内快速找到匹配串.

本文采用的搜索范围为全帧范围. 传统的哈希

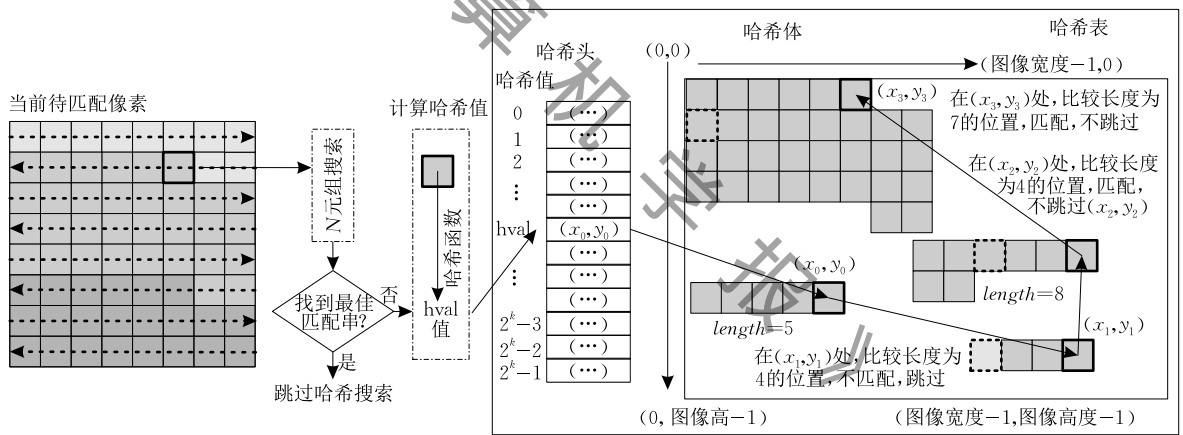


图 4 一种快速的哈希表搜索方案图示

如图 4 所示,对于当前待匹配像素,首先进行  $N$  元组搜索,若找到最佳参考串,则跳过哈希搜索,否则,进行快速哈希搜索. 例如,在哈希节点  $(x_0, y_0)$  处,找到了长度为 5 的参考串,根据最佳参考串评价规则  $R$ ,下一个参考串的长度必须最小为 4,才可能成为参考串,因此利用该性质,在下一个哈希节点  $(x_1, y_1)$  处检查从  $(x_1, y_1)$  位置长度为 4 的像素所在位置(用虚框矩形表示)是否满足匹配准则  $C$ ,不满足  $C$ ,则当前哈希节点不可能成为最佳参考串,直接跳过该哈希节点.

实验结果表明,在编码效率几乎不变的前提下,本文提出的快速哈希表搜索方案,编码时间复杂度降低 2% 左右.

表搜索算法<sup>[15]</sup>,如图 4 所示,对于当前待匹配像素,计算其哈希值  $hval$ ,根据哈希值在哈希头中找到第一个参考串的位置  $(x_0, y_0)$ ,逐像素进行匹配;沿着哈希链找到下一个参考串的位置  $(x_i, y_i)$ ,逐像素进行匹配,直至到达哈希链的末尾或者最大的搜索尝试次数(本文设置为 1000 次). 根据最佳参考串评价规则  $R^{opt}$  从中选出最佳参考串. 为了解决哈希表搜索算法对于一些不可能成为参考串的参考位置进行逐像素搜索的问题,提出了一种快速哈希表搜索方案. 该方案主要包括两个步骤:

(1) 首先,进行基于  $N$  元组的搜索,即仅对已经编码的最近串的  $N$  个位移矢量进行搜索,若能找到最佳参考串,则跳过哈希表搜索;

(2) 然后,对一个哈希节点进行传统的逐像素匹配之前,利用最佳参考串评价规则  $R^{opt}$  快速判断当前哈希节点的参考位置获得的参考串是否可能成为匹配串,如果不可能成为匹配串,则跳过该参考位置.

#### 4.3 改进的匹配关系参数编码方案

串匹配算法要大幅提高屏幕图像的编码效率,一个核心的问题是如何对匹配关系参数进行有效的编码. 对此,本文提出了基于分段截断二进制码长度参数编码方案和基于  $N$  元组的位移参数编码方案.

##### 4.3.1 基于分段截断二进制码的长度参数编码方案

设  $N(N=2^{k+1})$  个符号是取值  $1 \sim N$  的数,如果已知需要表示的符号有  $M$  个,而  $N/2 < M < N$ ,那么用  $k$  bits 来表示  $M$  个符号就不够而  $(k+1)$  bits 又太多. 截断二进制码(Truncated Binary Code, TB)<sup>①</sup>

① [https://en.wikipedia.org/w/index.php?title=Truncated\\_binary\\_encoding&oldid=973658](https://en.wikipedia.org/w/index.php?title=Truncated_binary_encoding&oldid=973658)



提供了一种解决方案:用  $k$  bits 的固定字长 2 进制码来表示  $M$  个符号中的  $J$  个符号而用  $(k+1)$  bits 的固定字长 2 进制码来表示  $M$  个符号中的其余  $M-J$  个符号.  $J$  称为截断值. 采用 TB 后,对  $M$  个符号中的  $J$  个符号,编码所需比特数可减少一个比特.

设当前编码 CU 的总像素数目为  $Mnum$ ,已编码(或解码串)的长度为  $Cnum$ ,剩下像素的数目为  $Max=Mnum-Cnum$ ,则当前串的长度必满足  $length$

小于等于  $Max$ . 利用  $length$  与  $Max$  的相关性,在分段定长编码方案的基础上<sup>[11]</sup>(见表 2),提出了长度参数分段 TB 编码方案,见表 3.

表 2 基于分段定长二进制的长度编码方案

长度(L)	前缀	后缀
0~3	1	FL(L,2)
4~19	01	FL(L-4,4)
20~275	001	FL(L-20,8)
276~4095	000	FL(L-276,12)

表 3 基于分段截断二进制的长度编码方案

长度(L)	前缀		后缀	
	$Max \leq 3$	$Max > 3$	$Max \leq 3$	$Max > 3$
0~3	无	1	TB(L, Max+1)	FL(L, 2)
4~19	$Max \leq 20$ 0	$Max > 20$ 01	TB(L-4, Max-3)	FL(L-4, 4)
20~275	$Max \leq 275$ 00	$Max > 275$ 001	TB(L-20, Max-19)	FL(L-20, 8)
276~4095	$Max \leq 4095$ 000	$Max > 4095$ 000	TB(L-276, Max-275)	FL(L-276, 12)

其中  $FL(synVal, k)$  表示对数值  $synVal$  用固定长度  $k$  bits 进行编码,  $TB(synVal, cMax)$  表示对数值  $synVal$  进行最大值为  $cMax$  的 TB 编码. 实验结果表明,提出的分段 TB 编码方案与分段编码方案相比,在 AI 配置下,对于 AVS2 SCC 通用测试序列<sup>[22]</sup>,Y-BD rate 平均降低 0.9%.

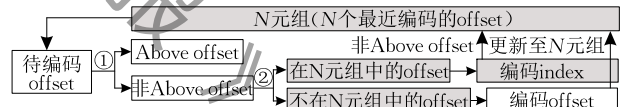
#### 4.3.2 基于 $N$ 元组的位移参数编码方案

为了提高位移参数的编码效率,文献[16]提出了一种高效的位移参数熵编码方案.该方案主要包括以下三部分的改进工作:根据 Offset 联合概率分布特性和 Offset 参数之间的相关性,提出了一种联合 Offset 编码方案——对频率出现最高的 ABOVE 分配最短的二值化码、Offset 绝对值采用 2 阶指数哥伦布编码、不同扫描方式时采用不同的  $OffsetYX$  二值化方案;根据 Offset 参数与其它复制参数  $length$  等之间的相关性,提出了水平扫描时  $OffsetX$  映射方案;提出了一种截断(Truncated Binary Code, TB)EG2 编码方案,根据当前编码位置与 Offset 的取值范围之间的相关性,将其应用到 Offset 后缀编码中.

该方案未考虑最近使用过(编解码顺序上邻近)的 Offset 经常会重复出现的特性,在文献[16]中的位移参数编码方案的基础上,提出了一种基于  $N$  元组的位移参数编码方案,进一步提高了编码效率.

如图 5 所示,对待编码的 offset,首先分成两类,

分别是 Above offset(行序优先扫描下, Above offset 为  $offsetX$  取值为 0,  $offsetY$  取值为 1;列序优先扫描下, Above offset 为  $offsetX$  取值为 1,  $offsetY$  取值为 0)和非 Above offset;对非 Above offset,保存  $N$  个最近串编码的非 Above offset(即  $N$  元组),如果在  $N$  元组中,则编码  $N$  元组中相对应位置的 index,否则按照文献[16]对非 Above offset 的编码方式进行编码.



① 表示编码是否是Above offset标识 ② 表示编码是否在N元组中标识

图 5 基于  $N$  元组的 offset 参数编码方案图示

基于  $N$  元组的建立与更新算法如下:

设  $R_1, R_2, \dots, R_N$  表示  $N$  元组中最近、第二最近、...、第  $N$  最近串的 Offset,在对一帧图像进行编码前将  $N$  元组所有的 Offset 都初始化为值  $(0, 0)$ . 设  $X$  是当前待编码 Offset,若  $X$  是非 Above 的 Offset 并且不等于  $N$  元组中任一个 Offset 的值,则从  $N$  到 2 依次将第  $n-1$  最近串放置在第  $n$  最近串的位置,将  $X$  放置在最近串的位置.

实验结果表明,基于  $N$  元组的位移参数编码方案与文献[16]的位移参数编码方案相比,在 AI 配置下,对于 AVS2 SCC 通用测试序列<sup>[22]</sup>,Y 分量 BD rate 平均降低 3.30%.

### 5 实验结果与分析

#### 5.1 实验测试序列

本文采用的测试数据集来自屏幕与混合内容视频编解码通用测试条件<sup>[22]</sup>中的通用测试序列. 使用 13 个 YUV 4:4:4 色彩格式, 每个分量 8bits 的测试序列. 表 4 给出了 AVS2 SCC 通用测试序列. 表 5 给出了通用测试数据集的特征. 13 个序列分为以下 3 个类别:

(1) 移动的文字和图形 (Text and Graphics with Motion, 简称 TGM) 类别, 含 8 个序列: Flying Graphics (FLYG)、Desktop (DSK)、Console (CNS)、ClearType Spreadsheet (SPS)、Bitstream Analyzer (BSA)、English Document Editing (EDE)、Chinese Document Editing (CDE)、Circuit Layout Presentation (CLP);

(2) 含自然视频或相片般逼真、不规则纹理、或

表 4 AVS2 SCC 通用测试序列

类别	短名	分辨率	帧率	编码帧数
TGM	FLYG	1920×1080	60	300
TGM	DSK	1920×1080	60	600
TGM	CNS	1920×1080	60	600
TGM	SPS	1920×1080	30	300
TGM	BSA	1920×1080	30	300
TGM	EDE	1920×1080	30	300
TGM	CDE	1920×1080	30	300
TGM	CLP	1920×1080	30	300
MC	PVS	1920×1080	30	300
MC	WEBB	1280×720	30	300
MC	MAP	1280×720	60	600
MC	PRG	1280×720	60	600
G	ROB	1280×720	30	300

表 5 AVS2 SCC 通用测试数据集特征

类别	短名	特征
TGM	FLYG	各种快速飞行, 向不同方向移动和旋转的图样
TGM	DSK	计算机日常应用中常见的文档编辑窗口、命令行窗口、移动的曲线窗口
TGM	CNS	文字快速滚动的命令行窗口、移动的条形图和曲线图等
TGM	SPS	ClearType 技术显示的阿拉伯数字, 使用 Excel 表格的典型场景
TGM	BSA	码流分析软件使用场景: 原图像, CU 划分线、不同颜色的 CU 模式等多图层半透明混合
TGM	EDE	ClearType 技术显示的英文的 ppt 文档和 word 文档的编辑, ppt 有常用的图案背景
TGM	CDE	ClearType 技术显示的中文的 ppt 文档和 word 文档的编辑, ppt 有常用的图案背景
TGM	CLP	多种集成电路和 PCB 板设计版图, 单像素直线, 拖曳的窗口轨迹的半透明混合叠加
MC	PVS	视频流播放窗口与 ClearType 技术显示的 Excel 文档窗口的混合
MC	WEBB	浏览网站和网页的典型场景, 网页上混合了文字和自然图片
MC	MAP	放大、缩小地图, 搜索、定位街道和位置, 混合了相片般逼真和不规则纹理的内容
MC	PRG	软件编程和调试的典型应用场景, 各种编辑、调试和运行窗口, 有渐变亮度内容的窗口
G	ROB	典型的动画游戏序列

渐变亮度场景的混合内容 (Mixed Content with Natural Video or Photorealistic, Irregular Texture-Mapped, Light-Shaded Scene, 简称 MC) 类别, 含 4 个序列: People In Vehicle Spreadsheet (PVS)、Web Browsing (WEBB)、Map (MAP)、Programming (PRG);

(3) 游戏 (Gaming, 简称 G) 类别, 含 1 个序列: Robot (ROB).

图 6 是两个代表性测试序列的第 1 帧.

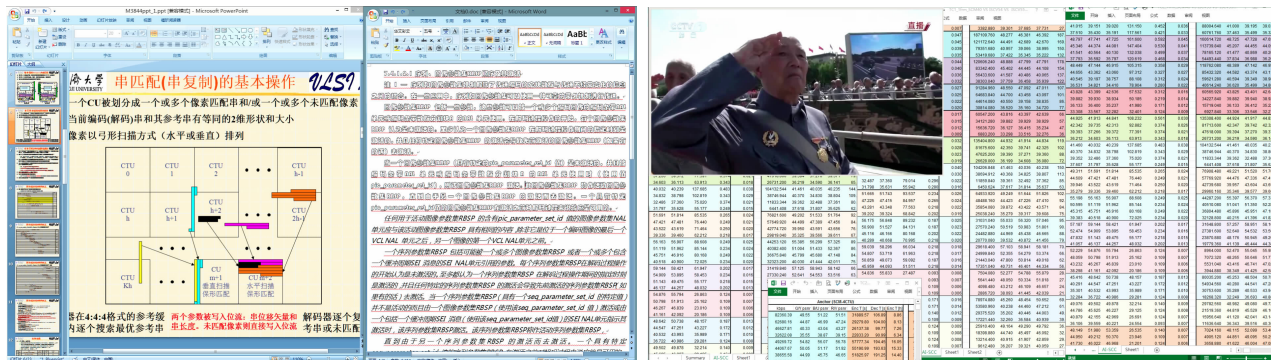


图 6 两个 AVS2 SCC 代表测试序列的第 1 帧图像

#### 5.2 实验配置与性能评价指标

为了便于说明本文算法的特性, 将本文提出的算法及其比较的基准分为以下 5 种算法:

(1) RD14.2Pseudo444. 在 AVS2-P2 基准档次

的参考软件 RD14.2 上增加对输入原始图像的色度下采样和对输出重构图像的色度上采样, 以实现 YUV4:4:4 序列的编码和解码, 相当于图 2 中下方的虚线方框;

(2) RD14.2SM. 在 RD14.2Pseudo444 上集成了像素串匹配算法, 以实现融合像素串匹配的全色度屏幕混合内容的编解码, 相当于图 2 的整体框架图. 像素串匹配算法同文献[13]中 PRB 像素串匹配算法.

(3) RD14.2ESM. 在 RD14.2SM 基础上的改进版本, 包括不同模式 RDcost 计算方式的优化和基于  $N$  元组的搜索与编码方案.

(4) HM16.7. HEVC 4:4:4 扩展版的参考软件<sup>①</sup>.

(5) SCM6.0. 基于 HM16.7 的 HEVC SCC 的参考软件<sup>②</sup>.

编码参数配置采用 AVS2 屏幕与混合内容通用测试条件中的标准配置<sup>[22]</sup>. 如表 6 所示, 采用有损编码模式, 有损编码模式的四个 QP 值 HEVC 为 {22, 27, 32, 37}, AVS2 为 {25, 32, 38, 45}, 使用 All Intra (AI, 即全部图像都是 I 图像) 和 Low Delay P (LDP, 即第一幅图像是 I 图像, 其余都是 F 图像) 两种编码图像类型配置. SM 的参考像素范围为当前 LCU 内.

实验中编码的帧数为 30 帧.

表 6 SCC 通用测试条件

编码精度模式	有损编码模式 Lossy
编码配置	(1) 全帧内 (All Intra, AI) (2) 低延迟 (Low Delay P, LDP)
编码数据色彩格式	YUV 4:4:4
SM 的参考像素范围也称参考范围或搜索范围	当前编解码的 LCU、LCU 及其左边三个 LCU (共四个 LCU) 或全帧范围 (本文采用当前 LCU 内)
QP 值	Lossy: 22, 27, 32, 37 和 25, 32, 38, 45

实验结果从编码效率和复杂度来衡量算法的性能. BD-rate<sup>[23-24]</sup> 是 HEVC 核心实验采用的比较两

种算法有损编码模式编码效率优劣的客观评价方法; 根据两种算法 4 个 QP 的比特率 (Bit-rate, 单位为 kbps) 与峰值信噪比 (Peak Signal to Noise Ratio, PSNR) 数据形成的率失真 (Rate Distortion, RD) 曲线, 对其进行曲线拟合, 计算两种算法在相同 PSNR 下比特率的变化情况, 最后取平均值得到两种算法编码效率的比较结果. BD-rate 为负值表示相同 PSNR 下比特率降低, 编码效率提高.

采用 Y、U、V 分量平均 BD-rate 降低率来衡量算法的总体编码效率. 采用 RD (Rate Distortion) 曲线图来衡量单个序列的编码效率. 算法复杂度的比较分别用国际标准制定工作中通用的编、解码运行时间的比值来衡量.

每种算法的 AI 配置在 CPU 为 Intel(R)@2.3 GHz 10 核, 内存为 40 GB 的服务器上执行; LDP 配置在 CPU 为 Intel(R)@2.3 GHz 20 核, 内存为 80 GB 的服务器上执行. 每次同时执行 13 个窗口. 操作系统为 Windows 7 64 位. 编解码时间测量不是很精确.

### 5.3 实验结果

表 7 给出了基于 AVS2 的 RD14.2Pseudo444、RD14.2SM、RD14.2ESM 三种算法总体编码效率与复杂度的两组 (RD14.2Pseudo444 相对 RD14.2SM、RD14.2SM 相对 RD14.2ESM) 比较结果. 编码效率比较中, 每行数据表示一个序列类别 (每个类别由 1~8 个序列组成, 见表 4), 分别给出了 AI、LDP 配置分量 Y、U、V 的平均 BD-rate 降低率. BD-rate 降低率表示在相同的峰值信噪比 (Peak Signal to Noise Ratio, PSNR) 取值下比特率节省的百分比. 负值表示比特率降低, 编码效率提高. 表 7 也给出了编解码运行时间的比值.

表 7 RD14.2Pseudo444、RD14.2SM、RD14.2ESM 算法编码效率与复杂度的两组比较结果

对比算法	测试序列类别	编码效率比较/%						复杂度比较/%		
		AI 配置			LDP 配置			AI 配置	LDP 配置	
		Y	U	V	Y	U	V			
RD14.2 PSEUDO444 VS. RD14.2ESM	TGM MC G	-56.1 -31.3 0.7	-97.2 -69.1 -5.4	-99.2 -81.3 -5.4	-41.8 -25.1 1.5	-70.9 -37.7 0.0	-72.9 -39.5 0.0	编码时间比值 解码时间比值	123.9 79.8	107.6 172.3
RD14.2SM VS. RD14.2ESM	TGM MC G	-16.6 -14.0 0.7	-17.2 -33.9 -4.9	-16.4 -36.7 -5.4	-7.9 -10.2 1.5	-26.5 -20.1 -1.2	-25.8 -20.7 0.3	编码时间比值 解码时间比值	100.9 87.1	101.3 99.8

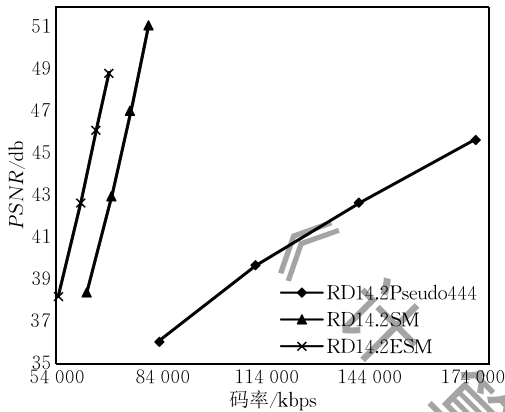
表 8 给出了基于 AVS2 的 RD14.2Pseudo444、RD14.2ESM 和基于 HEVC 的 HM16.7、SCM6.0 四种算法 AI 配置编码效率与复杂度的两组 (RD14.2Pseudo444 相对 RD14.2ESM、HM16.7 相

对 SCM6.0) 比较结果.

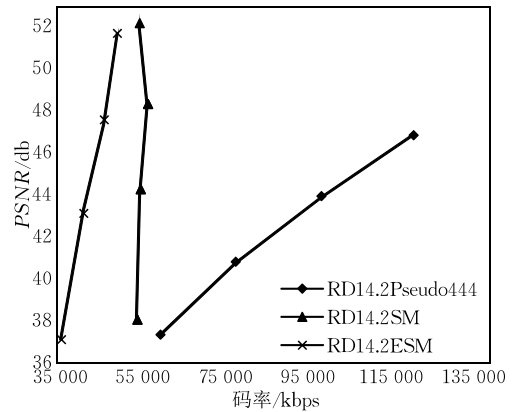
① [https://hevc.hhi.fraunhofer.de/svn/svn\\_HEVCSoftware/tags/](https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/)  
② [https://hevc.hhi.fraunhofer.de/svn/svn\\_HEVCSoftware/tags/HM-16.7+SCM-6.0/](https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.7+SCM-6.0/)

表 8 四种算法编码效率与复杂度的两组比较结果

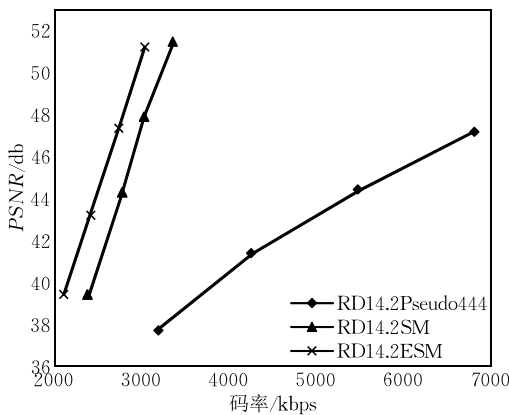
对比算法	测试序列类别	编码效率比较/%			复杂度比较/%	
		AI 配置			AI 配置	
		Y	U	V	编码时间比值	解码时间比值
RD14.2 PSEUDO444 VS. RD14.2ESM	TGM	-56.1	-97.2	-99.2	123.9	
	MC	-31.3	-69.1	-81.3		79.8
	G	0.7	-5.4	-5.4		
HM16.7 VS. SCM6.0	TGM	-76.8	-78.3	-77.9	346.9	
	MC	-61.3	-65.6	-69.1		61.7
	G	-2.5	-16.1	-12.6		



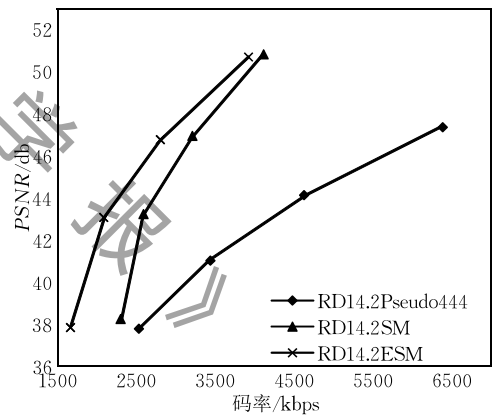
(a) CDE序列AI配置Y分量RD曲线



(b) PVS序列AI配置Y分量RD曲线



(c) CDE序列LDP配置Y分量RD曲线



(d) PVS序列LDP配置Y分量RD曲线

图 7 典型序列有损编码 RD 曲线比较

BD-rate 降低率为 56.1%、97.2%、99.2%，LDP 配置下 Y、U、V 分量平均 BD-rate 降低率为 41.8%、70.9%、72.9%。同时，RD14.2ESM 较 RD14.2SM 平均编码效率明显提升。对于 TGM 序列类别，AI 配置下 Y、U、V 分量平均 BD-rate 降低率为 16.6%、17.2%、16.4%。图 7 中典型序列的 RD 曲线同样表明 RD14.2ESM 的有效性。

(2) 从复杂度来看，对于 AI 配置和 LDP 配置，RD14.2ESM 较 RD14.2Pseudo444 编码复杂度分别增加了约 23.9%、7.6%。编码时间增加一方面在于 SM 编码器中进行最优匹配串搜索；另一方

面，AVS2 的帧内与帧间算法是基于 4:2:0 格式的，处理的 UV 分量的像素值数目是 SM 算法的 1/4。解码复杂度 AI 配置降低了 21.2%，LDP 配置增加了 72.3%（LDP 配置除了第一帧外都是 P 图像，RD14.2Pseudo444 中大多数 CU 选中 4:2:0 帧间 skip 模式，其解码复杂度很低，相对来说，RD14.2ESM 中 4:4:4 串匹配需要处理的 UV 分量数目增加四倍，总像素分量数目增加一倍，导致解码时间增加）。RD14.2ESM 较 RD14.2SM，编码复杂度几乎不变，解码时间稍有降低。

#### 5.4 实验结果分析

从以上实验结果，可以得出以下结论：

(1) 从编码效率来看，RD14.2ESM 较 RD14.2Pseudo444 平均编码效率大幅提升。从表 7 中可知，RD14.2ESM 与 RD14.2Pseudo444 算法相比，对于 TGM 序列类别，AI 配置下 Y、U、V 分量平均

面，AVS2 的帧内与帧间算法是基于 4:2:0 格式的，处理的 UV 分量的像素值数目是 SM 算法的 1/4。解码复杂度 AI 配置降低了 21.2%，LDP 配置增加了 72.3%（LDP 配置除了第一帧外都是 P 图像，RD14.2Pseudo444 中大多数 CU 选中 4:2:0 帧间 skip 模式，其解码复杂度很低，相对来说，RD14.2ESM 中 4:4:4 串匹配需要处理的 UV 分量数目增加四倍，总像素分量数目增加一倍，导致解码时间增加）。RD14.2ESM 较 RD14.2SM，编码复杂度几乎不变，解码时间稍有降低。

(3) 表 8 中 RD14.2ESM 与 RD14.2Pseudo444

的比较结果和 SCM6.0 与 HM16.7 的比较结果表明, RD14.2ESM 取得的编码效率提升与 HEVC SCC 取得的编码效率提升相比, 有的类别和分量更大, 而有的类别和分量更小, 总体差别不大, 但是前者编码复杂度仅增加了 23.9%, 远远低于 HEVC SCC 增加的编码复杂度(约为 246.9%)。

(4) 提出的 RD14.2SM 与 RD14.2ESM 算法性能的提升程度与屏幕图像类别相关。从表 7 中可以看出, 对于 TGM 类别, 提升比较多, M 类别有一些提升, 而对于 G 类别几乎没有提升, LDP 配置稍微有点降低。主要原因包括: ① 有的屏幕图像(如计算机产生的文字和图形部分)中存在很多重复的匹配图样, 而有的屏幕图像(如自然图像或视频部分)中存在很少重复的匹配图样; ② 增加的 SM 算法是一种帧内编码工具, 对于有些序列, SM 算法在 LDP 配置中几乎没有作用, 反而增加了表示 CU 编码类型的标志位比特数, 导致性能有些下降。事实上, 在云计算平台和远程桌面等人机互动的屏幕内容编码的实时应用中, AI 配置的性能决定了数据传送的瓶颈<sup>[1]</sup>, 远比 LDP 配置的性能重要。

## 6 结 论

在 AVS2-P2 框架中, 提出了一种低复杂度、高效的融合串匹配算法的全色度屏幕混合内容的编码算法。通过不同模式采用不同的 RDcost 计算方式解决基于 4:2:0 格式的帧内、帧间算法与基于 4:4:4 格式的 SM 算法的无缝衔接问题, 并且进一步改进了串匹配算法的性能。提出的算法能大幅提高屏幕混合内容的编码效率, 尤其对 TGM 类别特别有效。与 HEVC 中 SCC 扩展版相比, 提出的算法具有兼顾编码性能和编码复杂度、编码复杂度增加很小的优势。

下一步的工作包括在 AVS2-P2 中探索各种受限的 SM 算法及其与 AVS2-P2 基准档次中的现有帧内、帧间编码模式的兼容性问题。

## 参 考 文 献

- [1] Lin Tao, Zhou Kai-Lun, Wang Shu-Hui. Cloudlet-screen computing: A client-server architecture with top graphics performance. *International Journal of Ad Hoc and Ubiquitous Computing*, 2013, 13(2): 96-108
- [2] Xu Ji-Zheng, Joshi R, Cohen R A. Overview of the emerging HEVC screen content coding extension. *IEEE Transactions on Circuits and Systems for Video Technology*, 2016, 26(1): 50-62
- [3] Shen Yan-Fei, Li Jin-Tao, Zhu Zhen-Min, Zhang Yong-Dong. High efficiency video coding. *Chinese Journal of Computers*, 2013, 36(11): 2340-2355(in Chinese) (沈燕飞, 李锦涛, 朱珍民, 张勇东. 高效视频编码. *计算机学报*, 2013, 36(11): 2340-2355)
- [4] Budagavi M, Kwon D-K. Intra motion compensation and entropy coding improvements for HEVC screen content coding//*Proceedings of the Picture Coding Symposium*. San Jose, USA, 2013: 365-368
- [5] Guo Li-Wei, Pu Wei, Zou Feng, et al. Color palette for screen content coding//*Proceedings of the IEEE International Conference on Image Process*. Pairs, France, 2013: 5556-5560
- [6] Chen C, Peng W. Intra line copy for HEVC screen content coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 2016, (99):1-12
- [7] Lin Tao, Zhang Pei-Jun, Wang Shu-Hui, et al. Mixed chroma sampling-rate high efficiency video coding for full-chroma screen content. *IEEE Transactions on Circuits and Systems for Video Technology*, 2013, 23(1): 173-185
- [8] Lin Tao, Zhou Kai-Lun, Chen Xian-Yi, et al. Arbitrary shape matching for screen content coding//*Proceedings of the IEEE Picture Coding Symposium*, San Jose, USA, 2013: 369-372
- [9] Zhu Wei-Jia, Xu Ji-Zheng, Ding Wen-Peng, et al. Adaptive LZMA-based coding for screen content//*Proceedings of the IEEE Picture Coding Symposium*. San Jose, USA, 2013: 373-376
- [10] Ma Zhan, Wang Wei, Xu Meng, et al. Advanced screen content coding using color table and index map. *IEEE Transactions on Image Processing*, 2014, 23(10): 4399-4412
- [11] Li Bin, Xu Ji-Zheng, Wu F. 1-D dictionary mode for screen content coding//*Proceedings of the Visual Communications and Image Processing*. Valletta, Malta, 2014: 189-192
- [12] Zhao Li-Ping, Chen Xian-Yi, Tin Tao. Mapping of intra string copy parameters for HEVC screen content coding//*Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*. Siem Reap, Cambodia, 2014: 1-10
- [13] Zhao Li-Ping, Tin Tao, Zhou Kai-Lun, et al. Pseudo 2D string matching technique for high efficiency screen content coding. *IEEE Transactions on Multimedia*, 2016, 18(3): 339-350
- [14] Zhao Li-Ping, Zhou Kai-Lun, Guo Jing, et al. Universal string matching for ultra-high quality and ultra-high efficiency SCC. *JVET Document B0048*. San Diego, USA, 2016: 1-4
- [15] Li Bin, Xu Ji-Zheng. CE10: result of test 1 and 2. *JCTVC Document S0083*, Strasbourg, France, 2014: 1-4
- [16] Zhao Li-Ping, Lin Tao, Zhou Kai-Lun. An efficient ISC offset parameter coding algorithm in screen content coding. *Chinese Journal of Computers*, 2017, 40(5): 1218-1228(in Chinese)

(赵利平, 林涛, 周开伦. 屏幕图像压缩中串复制位移参数的高效编码算法. 计算机学报, 2017, 40(5): 1218-1228)

- [17] Gao Wen, Ma Si-Wei. *Advanced Video Coding Systems*. Berlin, Germany: Springer, 2015
- [18] Wang Shan-She, Luo Fa-Lei, Ma Si-Wei. Overview of the second generation AVS video coding standard (AVS2). *ZTE Communications*, 2016, (1): 1-13
- [19] AVS N14175. AVS Document, Dalian, China, 2015 (in Chinese)  
(AVS N14175. 数字音视频编解码技术标准工作组文件, 大连, 中国, 2015)
- [20] Lin Tao, Zhao Li-Ping, Zhou Kai-Lun, et. al. Pixel string matching for screen content coding in AVS2-P2. AVS Document M3844, Beijing, China, 2015 (in Chinese)  
(林涛, 赵利平, 周开伦等. 在现有 AVS2-P2 编码框架中增加串匹配编码单元类型的屏幕混合内容编码技术. 数字音视频编解码技术标准工作组文件 M3844, 北京, 中国, 2015)
- [21] Lin Tao, Zhao Li-Ping, Zhou Kai-Lun, et al. Universal

string matching for screen and mixed content coding in AVS2-P2. AVS Document M3915, Hangzhou, China, 2016 (in Chinese)

- (林涛, 赵利平, 周开伦等. 在现有 AVS2-P2 框架中增加通用串预测编码单元类型的屏幕与混合内容编码技术. 数字音视频编解码技术标准工作组文件 M3915, 杭州, 中国, 2016)
- [22] AVS N2282. Common test condition for screen and mixed content coding in AVS2-P2. Hangzhou, China, 2016 (in Chinese)  
(AVS N2282. AVS2-P2 屏幕与混合内容视频编码 (S&MCVC) 通用测试条件, 杭州, 中国, 2016)
- [23] Bjøntegaard G. Calculation of average PSNR differences between RD-curves. ITU-T SG16 Q. 6 Document, VCEG-M33, 2001
- [24] Bjøntegaard G. Improvements of the BD-PSNR model. ITU-T SG16 Q. 6 Document, VCEG-A111, 2008



**ZHAO Li-Ping**, born in 1984, Ph. D. candidate, lecturer. Her research interests include video coding, screen content coding.

**ZHOU Kai-Lun**, born in 1977, Ph. D. candidate, lecturer. His research interests include video coding and VLSI architecture, design and verification.

**GUO Jing**, born in 1993, M. S. candidate. His research interest is screen content coding.

**CAI Wen-Ting**, born in 1990, M. S. candidate. Her research interests focus on screen content coding.

**LIN Tao**, born in 1958, Ph. D., professor, Ph. D. supervisor. His research interests include video coding, screen content coding.

**ZHU Rong**, born in 1973, Ph. D., professor. Her research interests focus on intelligent information processing.

## Background

Screen content, such as the computer and mobile display content, has recently emerged as a popular video type due to the fast rising demands for transporting or storing screen display in the form of video. This is partly driven by rapid advances in mobile, cloud and display technologies, which enable a variety of screen applications over various networks/links, such as wireless displays, second screen, desktop screen sharing and collaboration, cloud computing and gaming, PC-over-IP, etc.

Three major HEVC SCC tools developed in recent years are: Intra Block Copy (IBC), palette coding (PLT) and String Matching (SM).

From video coding point of view, the most significant and relevant characteristic of screen content is probably its

diversity and comprehensiveness. Therefore, although SCC has made significant progress in the past few years, continuous improvement of performance is still indispensable and very challenging. A pixel string matching technique is proposed in AVS2 SCC project to meet with the diversity and comprehensiveness of the mixed contents of the screen, and has been adopted into AVS2-P2 SMCC extension working draft.

This work is supported by the National Natural Science Foundation of China (Nos. 61871289, 61601200), the Natural Science Foundation of Zhejiang Province (Nos. LY15F020039, LY18F020021, LQ18F020006, LY19F020015) and the Natural Science Foundation of Shanghai Province (No. 18ZR1440600).