

一种新的用于屏幕图像编码的 HEVC 帧内模式

陈先义^{*①} 赵利平^{①②} 林涛^①

^①(同济大学超大规模集成电路研究所 上海 200092)

^②(嘉兴学院数理与信息工程学院 嘉兴 314000)

摘要: 由于传统编码方式对屏幕图像的编码效果不佳, 该文根据屏幕图像包含大量非连续色调内容的特点, 在 HEVC(High Efficiency Video Coding)基础上, 提出一种新的帧内编码模式称为帧内串匹配(Intra String Copy, ISC)。基本思想是在 HEVC 的编码单元(Coding Unit, CU)级别上, 引入字典编码工具: 编码时, 在一定长度的字典窗口内, 利用散列表, 对当前 CU 内的像素, 进行串搜索和匹配; 解码时, 根据像素串匹配的距离和匹配长度, 在重建缓存内复制相应位置像素重建当前 CU 像素。实验结果表明, 在编码复杂度增加很少的情况下, 对于典型的屏幕图像测试序列, 在全帧内(All Intra, AI), 随机接入(Random Access, RA), 低延迟(Low-delay B, LB)3 种配置下, 有损编码模式比 HEVC 分别节省码率 15.1%, 12.0%, 8.3%, 无损编码模式分别节省码率 23.3%, 14.9%, 11.6%。

关键词: 高效视频编码; 屏幕图像编码; 字典编码; 散列表

中图分类号: TN919.8

文献标识码: A

文章编号: 1009-5896(2015)11-2685-06

DOI: 10.11999/JEIT150261

A New HEVC Intra Mode for Screen Content Coding

Chen Xian-yi^① Zhao Li-ping^{①②} Lin Tao^①

^①(Institute of VLSI, Tongji University, Shanghai 200092, China)

^②(College of Mathematics, Physics and Information Engineering, Jiaxing University, Jiaxing 314000, China)

Abstract: Because of the poor effect of the traditional coding methods on the screen content coding, considering the screen content is rich in non-continuous tone content, a new intra coding mode based on High Efficiency Video Coding (HEVC), which is called Intra String Copy (ISC), is proposed. The basic idea is adopting the dictionary coding tool on the HEVC Coding Unit (CU) level. When encoding, the current CU pixels are searched and matched in a certain length dictionary window by using Hash table. When decoding, according to the pixels string matching distances and lengths, the current CU pixels in the reconstruction cache are restored by copying the corresponding position pixels. Experiment results show that with little coding complexity increase than HEVC, for typical screen content test sequences, ISC can achieve lossy coding bit-rate saving of 15.1%, 12.0%, 8.3% for All Intra (AI), Random Access (RA), and Low-delay B (LB) configurations, respectively, and lossless coding bit-rate saving of 23.3%, 14.9%, 11.6% for AI, RA, and LB configurations.

Key words: High Efficiency Video Coding (HEVC); Screen content coding; Dictionary coding; Hash table

1 引言

随着云计算、移动云计算、远程桌面和无线显示技术的发展, 如何在低码率下使屏幕图像在电脑屏幕、手机屏幕、电视屏幕和其它客户端上高质量地显示, 吸引了学术界和工业界的关注。现有的视

频编码标准比如 H.264/AVC, VC-1, AVS, HEVC (High Efficiency Video Coding), 都是对于包含大量连续色调内容的自然图像和视频有很高的压缩比, 但是对于包含大量非连续色调内容的屏幕图像压缩效果并不好。为提高屏幕图像编码(Screen Content Coding, SCC)性能, 国际电信联盟电信标准化部门(ITU-T)、国际标准化组织(ISO)和国际电工委员会(IEC)于 2014 年 1 月开始联合开发 HEVC 标准的 SCC 版本, 并已经采用了一些有利于屏幕图像编码的工具: 比如跳过变换直接量化(Transform Skip, TS)^[1]和 IBC(Intra Block Copy)^[2,3]。

典型的屏幕图像包含两类内容, 一类是计算机生成的非连续色调的内容, 往往包含大量的小而尖的位图结构如文本、菜单、图标、按钮、滑动条和

收稿日期: 2015-02-22; 改回日期: 2015-06-10; 网络出版: 2015-07-27

*通信作者: 陈先义 chenxy666@126.com

基金项目: 国家自然科学基金(61201226, 61271096); 上海市自然科学基金(12ZR1433800); 高等学校博士学科点专项科研基金(20130072110054)

Foundation Items: The National Natural Science Foundation of China (61201226, 61271096); Natural Science Foundation of Shanghai (12ZR1433800); Specialized Research Fund for the Doctoral Program of Higher Education (20130072110054)

网格，另一类是摄像机拍摄的包含大量连续色调的内容，例如电影/电视片段，自然图像视频。

传统的基于角度预测的帧内编码模式，无法对非连续色调的内容进行准确的预测，因而压缩性能不好。而字典编码工具比如 gzip 和 LZMA (Lempel-Ziv-Markov chain Algorithm) 却对非连续色调内容有很好的压缩性能^[4]。于是文献[5~7]提出将字典编码工具融合到传统的混合编码框架中的 DMC (Dual-coder Mixed Chroma-sampling-rate) 算法，实验显示取得很好的性能。但是，DMC 算法是在条带内，基于 HEVC 编码树单元(Coding Tree Unit, CTU, HEVC 中的基本编码单元)实现的，而 CTU 大小默认是 64×64，显然，适用字典编码的区域可能小于 64×64，并且 DMC 算法对 4:4:4 格式的数据进行了子采样，转变为 4:2:0 格式，不可避免地引入失真。为了进一步提升编码性能，本文将字典编码工具应用于 HEVC 的编码单元(Coding Unit, CU, 由 CTU 按照二叉树划分得到，大小 8×8~64×64)，使之成为 HEVC 的一种帧内模式，叫做 ISC (Intra String Copy)，能够与 HEVC 原有编码工具更加和谐地融合在一起。实验显示比仅仅应用于 CTU 的 DMC 算法，性能明显提高。

2 ISC 算法概述

包含 ISC 的 HEVC 编码框架^[8]如图 1 所示。按照 CU 自适应的方式，每个输入的 CU 经过帧间、帧内、IBC, ISC 编码后，通过 HEVC 的率失真优化选择机制，选取 RD(Rate Distortion)代价最小的编码模式，将编码结果放入码流。

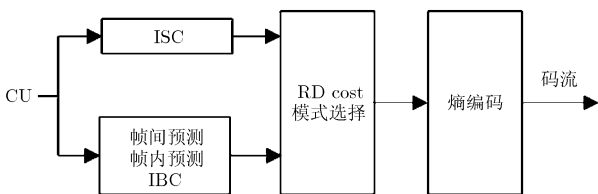


图 1 包含 ISC 的 HEVC 编码器框图

2.1 ISC 编码器和解码器

ISC 编码器和解码器如图 2 所示。

在编码端，输入的 CU 像素数据，首先经过像素重排器重新组织和排列。因为，在 HEVC 中，像素以 YUV 或 RGB 分量的 2 维平面格式组织，而 ISC 编码器实质上是 1 维字典编码器，其基于散列表的搜索需要在 1 维空间进行，于是需要通过一定的方式把 CU 像素数据重新排列，把 2 维数据 1 维化，同时把分离的 YUV 或 RGB 数据结合在一起，成为一个像素，以像素为单位存入搜索缓存。像素重排

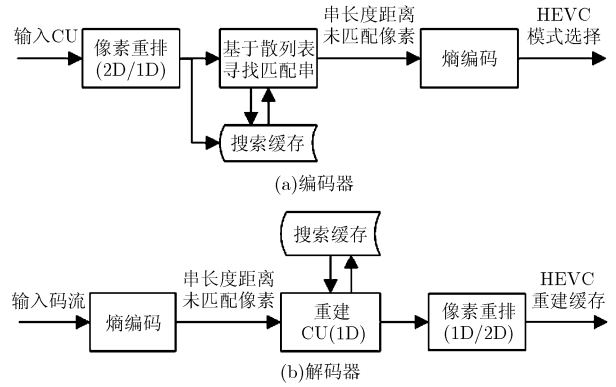


图 2 CU 自适应 ISC 编解码器架构

之后，在搜索缓存内，基于散列表寻找尽可能长的匹配串，串的长度最小值为 1，最大值为当前 CU 内像素总数。如果找到匹配串，保存串的长度和匹配点到当前位置的距离；如果没有找到匹配串，则保存当前像素，称为未匹配像素。然后在下一个位置像素继续搜索。整个 CU 搜索完成之后，对搜索的结果进行熵编码，得到 RD 代价，进行 HEVC 模式选择。如果确定采用 ISC 编码模式，需要在码流中写入一个比特的标识位。

在解码端，解码标识位，确定当前 CU 采用 ISC 编码之后，码流经过熵解码得到匹配串的长度、匹配距离、未匹配像素等句法元素。然后在搜索缓存内，根据这些句法元素重建当前 CU。即，如果有匹配，则根据匹配串的长度和距离，从搜索缓存内相应位置复制相应长度的像素到当前位置；如果没有匹配，则将解码得到的未匹配像素直接填充到当前位置。在搜索缓存内重建整个 CU 之后，需要把 1 维像素格式的重建数据，按照 HEVC 的 2 维平面格式，把 YUV 或 RGB 分量分别存入相应的 HEVC 重建缓存。

2.2 ISC 算法的相关句法

经由熵编码产生的 ISC 码流由以下句法元素组成：匹配标志(标识在搜索缓存中是否搜索到匹配串)，未匹配像素，距离(当前位置与搜索缓存中匹配位置之间的距离减 1)，长度(匹配串的长度减 1)，八元组标志(最近使用的 8 个距离值存放在一个数组中，称为八元组)，八元组索引(0~7)。对这些句法元素熵编码的解析过程如图 3 所示。

3 ISC 算法相关技术

3.1 搜索缓存

搜索缓存即字典，是 ISC 专用缓存，在编码器里用于匹配串搜索，在解码器中用于匹配串重建。在编码器，搜索缓存里存放当前正在编码 CU 的原

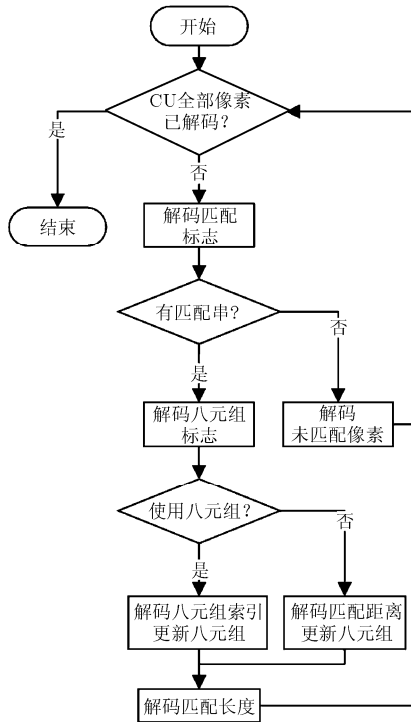


图 3 ISC 熵编码解析流程图

始像素和之前已经编码的 CU 的重建像素。搜索缓存的大小是预先指定的，有 7 个级别，Level1~7 分别为 4, 8, 16, 64, 256, 1024, 4096 kB。

显然，搜索缓存越大，越可能找到更优的匹配，但增加的时间和空间开销也就越多。实际最常用的缓存大小是 Level 4 的 64 kB。

3.2 像素重排

在 HEVC 中，像素以 3 个分量的平面格式组织，即 Y, U, V 或 R, G, B 3 个独立矩阵。但是这种像素组织方式并不适合 ISC 字典编码器。事实上，输入到 ISC 字典编码器的 CU 图像数据，首先在像素重排器内按照图 4 所示的某种扫描顺序，把 Y, U, V 或 R, G, B 3 个独立分量打包成一个像素 Pixel(YUV)或 Pixel(RGB)，以像素为单位，按照 1 维方式存放在搜索缓存中，即：Pixel₀(Y₀U₀V₀), Pixel₁(Y₁U₁V₁)... 或 Pixel₀(R₀G₀B₀), Pixel₁(R₁G₁B₁)...。可用的扫描顺序有图 4 所示的水平扫描、垂直扫描、水平旋转扫描和垂直旋转扫描 4 种。本文实际采用水平扫描方式。

3.3 基于散列表的搜索

ISC 的基本操作是在搜索缓存里，找到与当前要编码的像素相一致的像素，也就是寻找匹配串。如果在搜索范围和限定的搜索次数内能够找到匹配串，编码器输出匹配位置和当前位置之间的距离以及匹配串的长度；如果不能找到匹配串，编码器就

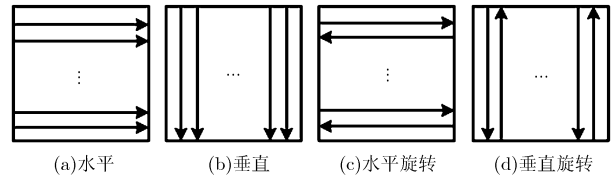


图 4 4 种扫描顺序

会输出当前像素，即未匹配像素。然后处理下一个像素。如果逐个像素地搜索整个缓存寻找匹配串，消耗的时间不可接受，为了加快搜索速度，有必要使用散列表：通过散列函数，将每个像素的 YUV 或 RGB 分量的 24 bit 映射为更少 bit 的散列值，依据散列值将像素在图像中的位置存入散列表，采用拉链开散列的方法处理散列冲突，即将相同散列值的像素位置放在同一条链上。搜索的时候，只需要计算当前像素的散列值，然后到对应的散列表的链上依次取得匹配位置，尝试匹配，搜索速度能够大大加快。散列函数如式(1)所示，其中 crc[] 是经过初始化的大小为 256 数组，Pix.Y, Pix.U, Pix.V 是像素的 Y, U, V 或 R, G, B 值，HashMask 为散列表表头大小减 1。

$$((\text{crc}[\text{Pix.Y}] \wedge \text{Pix.U}) \wedge (\text{Pix.V} \ll 8)) \& \text{HashMask} \quad (1)$$

由于每次编码完一个 CU，ISC 的搜索缓存指针（指示搜索位置）和散列表存储的内容都会发生变化，而 HEVC 对每一个 CTU 内所有 CU 的编码过程是先遍历二叉树的过程，于是在进行下一个 CU 编码之前，需要将搜索缓存指针和散列表表头恢复到一个正确的状态。可以按照如下步骤进行：

- (1) 每个 CU 在编码之前，首先保存搜索缓存指针和散列表表头。
- (2) 编码父 CU 的第 1 个子 CU 时，将搜索缓存指针和散列表恢复至父 CU 起始状态。
- (3) 编码其他子 CU 时，将搜索缓存指针和散列表恢复至这个 CU 的紧邻兄弟 CU 的起始状态，并读入这个兄弟 CU 的重构像素，用这些重构像素更新散列表。

散列表包括表头和链表两部分，其中链表与搜索缓存大小相同，随字典 Level 的变化而变化，在固定的字典 Level 下，表头的大小对 ISC 编码性能影响很小，对编码时间影响较大。表头太小，链表长度增加，搜索次数相应增加，增加编码时间；表头太大，由于表头需要频繁的保存和恢复，也会增加编码时间。实际选取 12 位的散列表位 (Hash Bits)，即散列表的表头大小为 2¹², 4k。

3.4 八元组

句法元素中的匹配距离，取值范围是(1~搜索

缓存大小),假设搜索缓存为Level 4,匹配距离的取值范围就是(1~16384)。显然,这是一个比较大的范围,为了更有效的编码匹配距离,引入了八元组,即:将最近出现的八个匹配距离存入数组,用索引(0~7)表示。在寻找匹配串的时候,首先在散列表里查找,记录下最长匹配串长度(mainLen)和匹配距离(mainDist);然后对八元组里保存的匹配距离尝试寻找匹配,如果找到的匹配串的长度(refLen)满足式(2),则认为八元组里寻找的匹配更优。

$$\begin{aligned} \text{refLen} &\geq \text{mainLen或}(\text{refLen}+1 \\ &= \text{mainLen且mainDist} \geq (1 \ll 14)) \quad (2) \end{aligned}$$

4 实验结果

为验证算法性能,以HEVC的SCC测试软件HM14.0_RExt7.0_SCM1.0^[9]为基础,实现ISC算法。所有实验以文献[10]规定的测试条件进行,分为无损(lossless)和有损(lossy)两种编码模式,每种模式有全帧内(All Intra, AI),随机接入(Random Access, RA),低延迟(Low-delay B, LB)3种配置。

实验选用的13个测试序列^[11]分为4类,分别为TGM(Text and Graphics with Motion),MC(Mixed Content),CC(Camera Captured)和A(Animation)。每个序列有RGB和YUV两种色彩格式,所以实际使用的测试序列有26个。测试序列的分辨率为720p,1080p或1440p,帧数为120~600不等。

为了评估算法性能,HEVC规定,对于无损编码,使用比特数变化表示码率变化,对于有损编码,使用BD-rate^[12,13]衡量同样客观质量下的码率变化。为了评估编码器和解码器复杂度,软件运行的编码时间和解码时间也进行比较。

实验比较了下述的3种编码方法:

(1)HM14.0_RExt7.0_SCM1.0(SCM)默认设置。

(2)HM14.0_RExt7.0_SCM1.0集成基于CTU的DMC算法(DMC)。

(3)HM14.0_RExt7.0_SCM1.0集成ISC算法(ISC)。

SCM中IBC的搜索范围是整幅图像,DMC和ISC均为64kB。

表1是ISC与SCM和DMC的编码、解码时间比较,其中数据为本文算法与对比算法时间总和的比值。表2列出了全部测试序列选中ISC模式的CU占图像面积的百分比。表3是ISC与SCM和DMC在无损和有损模式下的性能比较,每一行数据是一组色彩格式、类别、分辨率均相同的测试序列的平均值,所有的26个测试序列共分为12组,每

表1 ISC与SCM和DMC时间比较(%)

对比算法	时间比较	Lossless			Lossy		
		AI	RA	LB	AI	RA	LB
SCM	编码时间	112	105	105	114	100	99
	解码时间	83	89	89	98	104	105
DMC	编码时间	105	101	102	111	99	102
	解码时间	97	98	103	101	103	98

表2 ISC模式的CU占图像面积(%)

色彩格式	测试序列分组	Lossless			Lossy		
		AI	RA	LB	AI	RA	LB
R G B	TGM(1080p)	30.8	9.9	9.1	16.9	3.1	3.3
	TGM(720p)	36.3	5.5	4.0	8.7	1.7	1.3
	MC(1440p)	14.6	2.4	2.4	2.9	0.6	0.6
	MC(1080p)	13.5	1.4	1.4	3.1	0.4	0.3
	A(720p)	0.0	0.0	0.0	0.2	0.0	0.0
	CC(1080p)	0.0	0.0	0.0	0.0	0.0	0.0
Y U V	TGM(1080p)	32.5	9.5	8.4	16.4	2.5	2.1
	TGM(720p)	33.9	5.1	3.6	4.3	0.7	0.6
	MC(1440p)	12.7	2.0	2.0	1.7	0.4	0.4
	MC(1080p)	11.1	1.1	1.1	2.7	0.3	0.3
	A(720p)	0.0	0.0	0.0	0.1	0.0	0.0
	CC(1080p)	0.0	0.0	0.0	0.0	0.0	0.0
平均		21.2	4.3	3.6	6.4	1.1	1.0

组1~4个序列,负值表示本文算法与对比算法相比较节省了码率。图5给出了SCM,DMC,ISC3种编码器在有损编码模式,AI配置下,两个典型屏幕图像序列的率失真曲线。本文所列数据经过严格解码验证,ISC数据经过微软交叉验证^[14]。

从实验数据可以看出:

(1)无损模式下,本文ISC比SCM最大减少码率AI 23.3%,RA 14.9%,LB 11.6%;有损模式下,ISC比SCM最多节省码率15.1%。无损模式下,本文ISC算法比DMC最大减少码率AI 7.1%,RA 6.9%,LB 6.9%;有损模式下,ISC比DMC最多节省码率5%。

(2)对于不同类别的测试序列,ISC算法的性能不同。对于YUV TGM(1080p)这一组测试序列ISC算法有最好性能,比SCM无损模式平均提高23.3%,有损模式平均提高Y 13.9%,U 15.1%,V 14.3%;比DMC无损模式平均提高7.1%,有损模式平均提高Y 5.0%,U 5.0%,V 4.9%。对于其他的TGM以及MC类测试序列,ISC也有较好的性能增益。对于A

表 3 ISC 与 SCM 和 DMC 性能比较(%)

对比 算法	色彩 格式	测试序列 分组	Lossless			Lossy								
			AI	RA	LB	AI			RA			LB		
						G/Y	B/U	R/V	G/Y	B/U	R/V	G/Y	B/U	R/V
SCM	R G B	TGM(1080p)	-20.5	-12.6	-9.7	-13.6	-13.8	-13.8	-8.3	-8.8	-8.9	-4.9	-5.6	-5.6
		TGM(720p)	-10.3	-4.6	-3.3	-8.2	-7.5	-7.7	-8.6	-7.7	-8.0	-5.7	-5.2	-5.5
		MC(1440p)	-1.9	-0.2	-0.1	-1.5	-1.3	-1.3	-2.6	-2.1	-2.1	-2.2	-2.2	-2.0
		MC(1080p)	-2.2	-0.3	-0.1	-2.9	-2.9	-2.8	-3.3	-3.3	-3.1	-2.9	-2.8	-2.9
		A(720p)	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.1
		CC(1080p)	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.1	0.0	0.1	0.0
	Y U V	TGM(1080p)	-23.3	-14.9	-11.6	-13.9	-15.1	-14.3	-8.9	-11.0	-10.7	-4.6	-7.0	-6.7
		TGM(720p)	-10.3	-5.4	-3.7	-6.8	-9.0	-11.2	-7.5	-10.0	-12.0	-4.6	-6.9	-8.3
		MC(1440p)	-1.8	-0.2	-0.1	-1.0	-2.5	-2.4	-1.3	-4.0	-3.6	-1.3	-4.2	-4.2
		MC(1080p)	-1.8	-0.3	-0.1	-2.9	-5.2	-4.7	-3.1	-6.8	-6.0	-1.6	-6.7	-4.8
		A(720p)	0.0	0.0	0.0	0.1	0.0	0.0	0.1	-0.2	0.1	0.0	-0.1	0.1
		CC(1080p)	0.0	0.0	0.0	0.0	0.0	0.1	0.1	-0.1	0.1	0.1	0.1	0.0
DMC	R G B	TGM(1080p)	-6.5	-6.1	-6.1	-5.1	-5.0	-5.0	-4.2	-4.1	-4.2	-4.1	-4.1	-4.1
		TGM(720p)	-3.3	-2.8	-3.1	-3.5	-3.5	-3.4	-4.3	-3.9	-4.3	-3.9	-3.6	-4.0
		MC(1440p)	-1.3	-0.7	-0.7	-1.4	-1.3	-1.2	-1.9	-1.6	-1.7	-1.6	-1.4	-1.4
		MC(1080p)	-2.1	-0.8	-0.9	-1.8	-1.9	-1.5	-2.4	-2.4	-2.1	-2.1	-2.5	-1.9
		A(720p)	0.0	0.0	0.0	0.0	-0.2	0.0	0.0	0.2	0.1	0.2	0.0	0.1
		CC(1080p)	0.0	0.0	0.0	0.0	0.0	-0.1	-0.1	0.1	-0.1	0.1	0.0	-0.1
	Y U V	TGM(1080p)	-7.1	-6.9	-6.9	-5.0	-5.0	-4.9	-4.0	-4.1	-4.2	-3.8	-4.0	-4.0
		TGM(720p)	-3.4	-3.0	-3.1	-2.8	-3.4	-4.0	-3.0	-3.7	-4.6	-3.1	-3.7	-4.4
		MC(1440p)	-1.2	-0.7	-0.7	-0.7	-1.3	-1.2	-0.9	-1.8	-1.5	-0.7	-1.1	-1.3
		MC(1080p)	-1.9	-0.8	-0.8	-2.2	-2.9	-3.0	-2.2	-2.6	-2.4	-2.8	-3.7	-3.6
		A(720p)	0.0	0.0	0.0	0.0	-0.3	-0.1	-0.3	-0.4	0.3	-0.1	-0.9	0.1
		CC(1080p)	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.1	0.0	0.0	-0.3	0.0

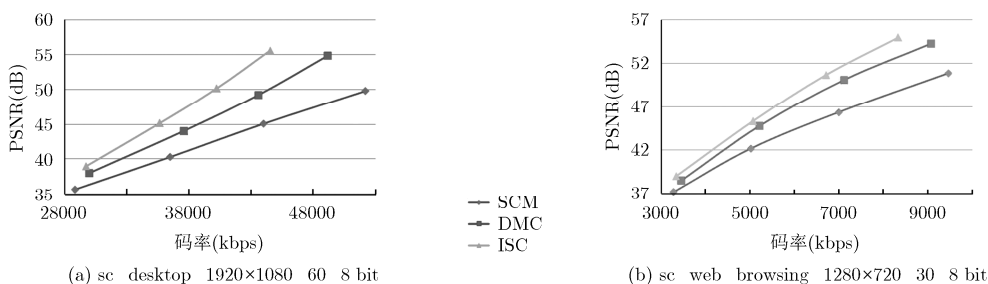


图 5 3 种编码器率失真(RD)曲线比较

类和 CC 类测试序列, ISC 没有增益。这是由于, A 和 CC 类序列非常近似于自然图像, 不是严格意义上的屏幕图像, 几乎所有屏幕图像相关算法均对这两类序列没有增益。

(3)ISC 算法对于无损编码性能增益大于有损编码。这是因为, ISC 算法本身是一种无损匹配, 搜

索过程中不会产生残差, 在无损编码情况下, 避免了残差熵编码, 较其他会产生残差的算法更有优势, 获得的性能增益更多。从表 2 可以看出, 选中 ISC 的 CU 的图像面积的占比, 全 I 帧, 无损最多有 36.3%, 平均 21.2%, 有损最多有 16.9%, 平均 6.4%, 无损模式高于有损模式, 并且对于 TGM 类序列选

中的面积明显高于其他类序列。这些都与 ISC 性能增益的数据保持一致,选中 ISC 的 CU 越多,ISC 的增益往往越大。

(4)从图 5 的率失真曲线可以看出,在高码率的情况下,ISC 的性能增益大于低码率的情况。这是因为,随着量化参数增大,码率降低,在率失真优化选择的时候,码率越来越重要,失真重要性相应降低,没有失真的 ISC 模式越来越难以在率失真优化选择中胜出。

(5)从表 1 编码时间来看,ISC 算法复杂度增加非常少。从解码时间来看,ISC 比 DMC 和 ISC 都有所减少,且 ISC 性能提升越多的序列,解码时间越短,这是由于 ISC 无需进行残差解码,重构过程比较简单。另外,ISC 算法的空间开销也十分有限,以 Level 4 为例,仅仅需要 64 kB 搜索缓存和 272 kB 散列表。

5 结束语

根据屏幕图像的特点,结合字典编码工具,在前人工作的基础上,本文提出了一种基于 HEVC 的 CU 自适应的处理屏幕图像的 ISC 帧内编码模式。在复杂度增加很有限的情况下,对屏幕图像的编码性能有显著改善。采用标准测试条件的实验显示,在无损编码模式下,AI, RA, LB 3 种配置分别比 HEVC 降低码率 23.3%, 14.9%, 11.6%, 有损编码模式下,分别比 HEVC 降低码率 15.1%, 12.0%, 8.3%。ISC 在无损编码方面有着更大的性能增益,这得益于 ISC 本身的无损匹配。

ISC 性能还可以进一步提升:一种方法是,对于有损编码模式,引入颜色量化的方法,使像素值的分布更加集中,减少整个搜索空间内像素的种类,进行模糊匹配,更容易找到长的匹配串。另一种方法是增加 CU 内像素自适应扫描顺序,即改变现在单一的水平扫描方式,增加垂直扫描,旋转扫描等其他方式,CU 自适应选择最优扫描方式,可以同时提升有损和无损编码性能。这些是下一步的研究方向。

参 考 文 献

- [1] Zhou M H, Gao W, Jiang M Q, *et al.* HEVC lossless coding and improvements[J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2012, 22(12): 1839-1843.
- [2] Budagavi M and Kwon D K. Video coding using intra motion compensation[C]. Joint Collaborative Team on Video Coding 13th Meeting, Incheon, 2013: JCTVC-M0350.
- [3] Pang C, Sole J, Guo L W, *et al.* Intra motion compensation with 2-d MVs[C]. Joint Collaborative Team on Video Coding

- 14th Meeting, Vienna, 2013: JCTVC-N0256.
- [4] Sullivan G and Ohm J R. Meeting report of the fifth meeting of the Joint Collaborative Team on Video Coding (JCT-VC) [C]. Joint Collaborative Team on Video Coding 5th Meeting, Geneva, 2011: JCTVC-E600.
- [5] Lin T, Zhang P J, Wang S H, *et al.* Mixed chroma Sampling-rate high efficiency video coding for full-chromascreen content[J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2013, 23(1): 173-185.
- [6] 张培君,王淑慧,周开伦,等.融合全色度 LZMA 与色度子采样 HEVC 的屏幕图像编码[J]. *电子与信息学报*, 2013, 35(1): 196-202.
Zhang P J, Wang S H, Zhou K L, *et al.* Screen content coding by combined full-chroma LZMA and subsampled-chroma HEVC[J]. *Journal of Electronics & Information Technology*, 2013, 35(1): 196-202.
- [7] Lin T, Wang S H, Zhang P J, *et al.* P2M based dual-coder extension of HEVC[C]. Joint Collaborative Team on Video Coding 12th Meeting, Geneva, 2013: JCTVC-L0303.
- [8] 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.
- [9] Bossen F, Flynn D, Sharman K, *et al.* Test Software [CP/OL]. https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-14.0+RExt-7.0+SCM-1.0, 2014.
- [10] Yu H P, Cohen R, Rapaka K, *et al.* Common conditions for screen content coding tests[C]. Joint Collaborative Team on Video Coding 17th Meeting, Valencia, 2014: JCTVC-Q1015.
- [11] Yu H, Cohen R, Rapaka K, *et al.* Test Sequences[DB/OL]. ftp://hevc@ftp.tnt.uni-hannover.de/testsequences/FrExt-candidate-sequences/screen_content/ScExt-TestSequences, 2014.
- [12] Bjøntegaard G. Calculation of average PSNR differences between RD-Curves[C]. Video Coding Experts Group 13th Meeting, Austin, 2001: VCEG-M33.
- [13] Bjøntegaard G. Improvements of the BD-PSNR model[C]. Video Coding Experts Group 35th Meeting, Berlin, 2008: VCEG-AI11.
- [14] Li B and Xu J Z. SCCE4: Cross-check of test 3.3[C]. Joint Collaborative Team on Video Coding 18th Meeting, Sapporo, 2014: JCTVC-R0060.

陈先义: 男, 1981 年生, 博士生, 研究方向为视频编码算法。
赵利平: 女, 1983 年生, 博士生, 研究方向为视频编码算法。
林 涛: 男, 1958 年生, 教授, 主要研究方向为多媒体算法和 SoC 设计。