

# HEVC 和 H.264 视频压缩实验报告

胡洪宇 517021910310

2019 年 12 月 1 日

## 1 实验一：HEVC 视频编码与解码

### 1.1 实验内容

参考代码 HM16.12 或以后版本。选择至少两个测试序列，具有不同空间分辨率、不同运动和纹理特性；对每个测试序列，分别设置两组不同的编码参数，进行编码和解码；给出相应的参数配置，给出若干关键帧的原始图像、以及相应的解码重建图像，给出每帧图像的 PSNR 值。

### 1.2 实验原理

HEVC 是 High Efficiency Video Coding 的缩写，是一种新的视频压缩标准，用来以替代 H.264/AVC 编码标准。

HEVC 使用了其他标准广泛应用的混合编码方法，即采用帧内、帧间预测和二维变换编码。HEVC 编码器首先将第一帧或随机存取点的某一 I 帧分割成多个块区域。当像素块仅依据当前帧的数据进行编码时，编码称作帧内编码。对于其他帧，使用了参考帧信息的编码方法称为帧间编码。预测运算和环路滤波结束后，重建图像储存于解码缓存中，可作为其他帧的参考帧。

为了应对不同应用场合，HEVC 设立了 GOP 的 3 种编码结构，即帧内编码、低延时编码和随机访问编码。在帧内编码结构中，每一帧图像都是按帧内方式进行空间域预测编码，不使用时间参考帧。在低时延 (Low Delay) 编码结构中，只有第一帧图像按照帧内方式进行编码，并成为 IDR 帧，随后的各帧都作为一般的 P 帧和 B 帧进行编码。

### 1.3 实验操作与分析

参考代码版本：HM-16.18；视频播放器：YUV Player；实验环境：Visual Studio2019；码流分析：Elecard HEVC Analyzer。

其中参考的两段视频为：“foreman-short-cif.yuv”每一帧大小为 352x288，内容为一个带安全帽的男人；CREW-352x288-15-orig-01.yuv”每一帧的大小为 352x288，内容为一段宇航员的视频。

### 1.3.1 序列一的第一种参数设定

对于第一个序列，使用 encoder-intra-main.cfg 作为编码参数设置文件，部分参数如下，该配置是一个全 I 帧的结构。

```
#===== Coding Structure =====
IntraPeriod      : 1      # Period of I-Frame ( -1 = only first)
DecodingRefreshType : 1      # Random Accesss 0:none, 1:CRA, 2:IDR, 3:Recovery
Point SEI
GOPSize          : 1      # GOP Size (number of B slice = GOPSize-1)
ReWriteParamSetsFlag : 1      # Write parameter sets with every IRAP
#      Type POC QPoffset QPfactor tcOffsetDiv2 betaOffsetDiv2 temporal_id
#ref_pics_active #ref_pics reference pictures
```

图 1: configuration1

同时，注意量化参数中 QP 参数设为 28. 如果设为 51，虽然能够节省码流，但是压缩太多，会导致视频质量很差。

视频压缩过程如下：

```
SUMMARY -----
      Total Frames |   Bitrate   Y-PSNR   U-PSNR   V-PSNR   YUV-PSNR
              30  a   1483.6640  38.0938  41.4026  44.1516  39.1363

I Slices-----
      Total Frames |   Bitrate   Y-PSNR   U-PSNR   V-PSNR   YUV-PSNR
              30  i   1483.6640  38.0938  41.4026  44.1516  39.1363

P Slices-----
      Total Frames |   Bitrate   Y-PSNR   U-PSNR   V-PSNR   YUV-PSNR
              0    p   -nan(ind) -nan(ind) -nan(ind) -nan(ind) -nan(ind)

B Slices-----
      Total Frames |   Bitrate   Y-PSNR   U-PSNR   V-PSNR   YUV-PSNR
              0    b   -nan(ind) -nan(ind) -nan(ind) -nan(ind) -nan(ind)

RVM: 0.000
Bytes written to file: 185458 (1483.664 kbps)

Total Time:      126.431 sec.
```

图 2: 序列一参数的一过程

整个编码过程一共花费 126.431s，一共压缩了 30 帧，其信噪比（PSNR）值为 39.1363。通过查看资料，当 PSNR 值为 40 以上时，认为视频（图像）是高质量的，PSNR 在 30 至

40 帧时是可以接受的。由此可见，从 PSNR 值反映出来的结果证明压缩过程是很成功的。

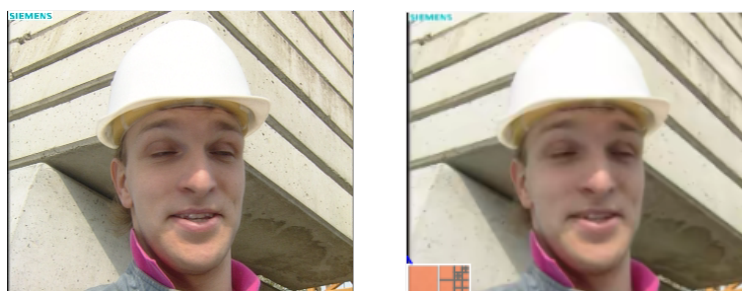


图 3: 第 1 帧 (I) 帧对比



图 4: 第 15 帧 (I) 帧对比



图 5: 第 30 帧 (I) 帧对比

由上述三帧关键帧的截取，对比原视频和编码解码之后的结果，可以发现：每一帧的图像和原始图像对比起来，总体质量相当，但是经过压缩之后还是有部分细节的损失，显得图片有些平滑和模糊。但是总体的效果还是算很好的。

虽然此时清晰度略有下降，但是可以看到，原视频大小为 4454KB，经过编码之后的文件仅为 182KB，整体压缩比为 24. 所以说明此时的压缩是很成功的。

rec.yuv	2019/11/30 13:47	YUV 文件	4,455 KB
str	2019/11/30 13:47	BIN 文件	182 KB

图 6: 文件大小对比

### 1.3.2 序列一的第二种参数设定

此时采取另一种参数设置方式 encoder-lowdelay-P-main, 这是一个第一帧为 I 帧、其余均为 P 帧编码方式。

```
#===== Coding Structure =====
IntraPeriod      : -1      # Period of I-Frame ( -1 = only first)
DecodingRefreshType : 0      # Random Accesss 0:none, 1:CRA, 2:IDR, 3:Recovery
Point SEI
GOPSize          : 4      # GOP Size (number of B slice = GOPSize-1)
ReWriteParamSetsFlag : 1      # Write parameter sets with every IRAP
```

图 7: configuration2

同时量化参数中 QP 参数采取默认的 32. 整个视频的压缩过程如下:

```
SUMMARY -----
Total Frames | Bitrate   Y-PSNR   U-PSNR   V-PSNR   YUV-PSNR
      30  a    133.0880  34.5039  40.1162  42.5810  35.7664

I Slices-----
Total Frames | Bitrate   Y-PSNR   U-PSNR   V-PSNR   YUV-PSNR
      1  i    958.0800  36.4618  40.7261  43.4939  37.6416

P Slices-----
Total Frames | Bitrate   Y-PSNR   U-PSNR   V-PSNR   YUV-PSNR
      29  p    104.6400  34.4364  40.0952  42.5496  35.7142

B Slices-----
Total Frames | Bitrate   Y-PSNR   U-PSNR   V-PSNR   YUV-PSNR
      0  b    -nan(ind) -nan(ind) -nan(ind) -nan(ind) -nan(ind)

RVM: 0.000
Bytes written to file: 16636 (133.088 kbps)

Total Time:      304.793 sec.
```

图 8: 序列一参数二的过程

整个过程花费时间 304.793s, 一共压缩了 30 帧。由此可以看到, 整体而言只有一帧为 I 帧, 其他都为 P 帧。整体的 PSNR 值为 35.7664. 该信噪比的结果也是可以接受的。关键帧截取的效果如下所示:



图 9: 第 1 帧 (I) 帧对比



图 10: 第 15 帧 (P) 帧对比



图 11: 第 30 帧 (P) 帧对比

编码后的压缩文件对比如下图所示：

rec.yuv	2019/11/30 14:28	YUV 文件	4,455 KB
str	2019/11/30 14:28	BIN 文件	17 KB

图 12: 文件大小对比

由此可以看见，此时的编码方式压缩比更高，更加节约空间。此时的压缩比可以达到 200 倍以上。

### 1.3.3 序列二的第一种参数设定

对第二个序列也采取相似的操作，第一种参数设定先只压缩 I 帧，采用 encoder-intra-main.cfg 进行编码压缩。具体配置和序列一的第一种参数设定相同。具体的运行结果如下：

SUMMARY							
Total	Frames		Bitrate	Y-PSNR	U-PSNR	V-PSNR	YUV-PSNR
30	a		1403.7600	39.1009	42.6949	41.7454	39.8925
I Slices							
Total	Frames		Bitrate	Y-PSNR	U-PSNR	V-PSNR	YUV-PSNR
30	i		1403.7600	39.1009	42.6949	41.7454	39.8925
P Slices							
Total	Frames		Bitrate	Y-PSNR	U-PSNR	V-PSNR	YUV-PSNR
0	p		-nan(ind)	-nan(ind)	-nan(ind)	-nan(ind)	-nan(ind)
B Slices							
Total	Frames		Bitrate	Y-PSNR	U-PSNR	V-PSNR	YUV-PSNR
0	b		-nan(ind)	-nan(ind)	-nan(ind)	-nan(ind)	-nan(ind)
RVM: 0.000							
Bytes written to file: 175470 (1403.760 kbps)							
Total Time: 123.840 sec.							

图 13: 序列二参数一的过程

整个过程花费了 126.431s，一共压缩了 30 帧。整体的 PSNR 值为 39.1363。该信噪比的结果可以接受。关键帧截取的效果如下所示：



图 14: 第 1 帧 (I) 帧对比



图 15: 第 15 帧 (I) 帧对比



图 16: 第 30 帧 (I) 帧对比

此时的压缩效果和 PSNR 值都在可接受范围内。整体压缩之后的两个文件对比，压缩比仍然在 25 左右。

rec.yuv	2019/11/30 14:57	YUV 文件	4,455 KB
str	2019/11/30 14:57	BIN 文件	172 KB

图 17: 文件大小对比

#### 1.3.4 序列二的第二种参数设定

对于第二种参数选择，仍然选择 encoder-lowdelay-P-main，这是一个第一帧为 I 帧、其余均为 P 帧编码方式。但是将压缩的参数调到了 28。



SUMMARY							
Total	Frames		Bitrate	Y-PSNR	U-PSNR	V-PSNR	YUV-PSNR
30	a		376.2640	36.7876	41.8028	40.3933	37.5997
I Slices							
Total	Frames		Bitrate	Y-PSNR	U-PSNR	V-PSNR	YUV-PSNR
1	i		1261.4400	40.1911	43.9764	43.3010	41.0645
P Slices							
Total	Frames		Bitrate	Y-PSNR	U-PSNR	V-PSNR	YUV-PSNR
29	p		345.7407	36.6702	41.7279	40.2930	37.5181
B Slices							
Total	Frames		Bitrate	Y-PSNR	U-PSNR	V-PSNR	YUV-PSNR
0	b		-nan(ind)	-nan(ind)	-nan(ind)	-nan(ind)	-nan(ind)
RVM: 0.000							
Bytes written to file: 47033 (376.264 kbps)							
Total Time: 456.247 sec.							

图 18: 序列二参数二的过程

整个过程花费了 456.247s，压缩编码了 30 帧图像。其 PSNR 值为 37.5997。该信噪比的结果可以接受。关键帧的截取效果如下。



图 19: 第 1 帧 (I) 帧对比



图 20: 第 15 帧 (P) 帧对比





图 21: 第 30 帧 (P) 帧对比

此时的压缩效果和 PSNR 值都在可接受范围内。整体压缩之后的两个文件对比，压缩比只有 10 左右。

rec.yuv	2019/11/30 16:03	YUV 文件	4,455 KB
str	2019/11/30 16:03	BIN 文件	46 KB

图 22: 文件大小对比

## 1.4 实验对比的分析

序列	参数选择	分辨率	编码帧数	编码前文件大小	编码后文件的大小	用时	PSNR 值
1	1	352x288	30	4455KB	182KB	126.431796s	39.1363dB
1	2	352x288	30	4455KB	17KB	304.793s	35.7664dB
2	1	352x288	30	4455KB	172KB	126.431s	39.1363dB
2	2	352x288	30	4455KB	46KB	456.247s	37.5997dB

表 1: 不同编码参数对压缩结果的影响

除此之外，考虑分析上述两个序列的两种参数设定的细节，还可以发现如下规律：

1、对比序列一的参数二和序列二的参数二。对于相同的压缩形式（I 帧和 P 帧的设定方式），量化参数中的 QP 设定的越低，压缩解码的效果越好，但是压缩比越低，压缩用时越长。这一点可以从。由此，视频的清晰度和压缩比是一对折衷的指标，在实际运用中还要根据实际情况来考虑。为此特意验证，取序列 的参数，QP 值分别取 1 和 51，可以得到如下结果：



图 23: 不同 QP 值的对比

其中  $QP=1$  时, 用时 600.451s, PSNR 达到了 60.212dB, 压缩比为 3;  $QP=51$  时, 用时 70.124s, PSNR 仅为 26dB, 压缩比为 2000。

2、对比第二个序列的两种不同的压缩方式, 可以发现, 第二种方式——这是一个第一帧为 I 帧、其余均为 P 帧编码方式比第一种方式——全为 I 帧的方式更花时间。

## 2 实验二: H.264 和 HEVC 视频编解码对比

### 2.1 实验内容

选择同一个测试序列, 采用相同编码参数, 分别进行 H.264 和 HEVC 压缩编码和解码; 对比分析两个不同编解码器的编码性能, 给出相应的参数配置, 给出若干关键帧的原始图像、以及相应的解码重建图像, 给出每帧图像的 PSNR 值。

### 2.2 实验原理

H.264, 同时也是 MPEG-4 第十部分, 是由 ITU-T 视频编码专家组 (VCEG) 和 ISO/IEC 动态图像专家组 (MPEG) 联合组成的联合视频组 (JVT, Joint Video Team) 提出的高度压缩数字视频编解码器标准。

### 2.3 实验操作与分析

仍然采用下列序列: “foreman-short-cif.yuv” 每一帧大小为 352x288, 内容为一个带安全帽的男人; CREW-352x288-15-orig-01.yuv” 每一帧的大小为 352x288, 内容为一段宇航员的视频。

#### 2.3.1 参数一——全 I 帧编码参数的对比分析

用 foreman-short-cif.yuv 序列进行 H.264 的编解码, 编码 30 帧, 其编码过程如下:

Frame	Bit/pic	WP	QP	SnrY	SnrU	SnrV	Time(ms)	MET(ms)	Frnm/Fld	I D
0000 (NVB)	176									
0000 (IDR)	62328	0	28	37.947	41.131	44.158	1009	0	FRM	396
0001 (I)	62344	0	28	37.960	41.185	44.332	1062	0	FRM	396
0002 (I)	62320	0	28	37.954	41.250	44.164	1024	0	FRM	396
0003 (I)	63120	0	28	37.963	41.272	44.363	1009	0	FRM	396
0004 (I)	63240	0	28	37.910	41.300	44.312	999	0	FRM	396
0005 (I)	63808	0	28	37.873	41.254	44.397	1078	0	FRM	396
0006 (I)	64400	0	28	37.872	41.223	44.192	1075	0	FRM	396
0007 (I)	64392	0	28	37.843	41.174	44.308	1015	0	FRM	396
0008 (I)	64264	0	28	37.865	41.154	44.121	1064	0	FRM	396
0009 (I)	64584	0	28	37.752	41.076	44.034	1063	0	FRM	396
0010 (I)	65640	0	28	37.745	41.181	43.809	1066	0	FRM	396
0011 (I)	65872	0	28	37.727	41.106	43.964	1034	0	FRM	396
0012 (I)	66712	0	28	37.769	41.181	44.038	1027	0	FRM	396
0013 (I)	67576	0	28	37.775	41.111	44.019	1057	0	FRM	396
0014 (I)	67128	0	28	37.695	41.023	43.955	1078	0	FRM	396
0015 (I)	67752	0	28	37.605	40.974	44.050	1106	0	FRM	396
0016 (I)	69112	0	28	37.500	40.993	44.184	1034	0	FRM	396
0017 (I)	69672	0	28	37.481	40.977	43.721	1035	0	FRM	396
0018 (I)	69160	0	28	37.396	40.944	44.133	1040	0	FRM	396
0019 (I)	69208	0	28	37.480	40.951	44.195	1035	0	FRM	396
0020 (I)	69120	0	28	37.464	41.013	44.093	1038	0	FRM	396
0021 (I)	68544	0	28	37.524	40.891	44.119	1058	0	FRM	396
0022 (I)	67792	0	28	37.574	40.869	44.116	1083	0	FRM	396
0023 (I)	67464	0	28	37.564	40.956	43.943	1031	0	FRM	396
0024 (I)	68344	0	28	37.525	41.032	44.040	1033	0	FRM	396
0025 (I)	70096	0	28	37.488	41.111	44.022	1028	0	FRM	396
0026 (I)	70688	0	28	37.429	41.107	44.010	1031	0	FRM	396
0027 (I)	70984	0	28	37.421	41.063	44.094	1036	0	FRM	396
0028 (I)	70408	0	28	37.379	41.052	44.079	1041	0	FRM	396
0029 (I)	70976	0	28	37.461	40.986	44.059	1056	0	FRM	396

图 24: H.264 编码过程

编码对应的结果如下,其中用时 33.675s,编码 30 帧,PSNR 值为  $SNR_Y = 37.66, SNR_U = 41.08, SNR_V = 44.10$

Average data all frames	
SNR Y(dB)	: 37.66
SNR U(dB)	: 41.08
SNR V(dB)	: 44.10
Total bits	: 2007224 (I 62328, P 1944720, NVB 176)
Bit rate (kbit/s) @ 30.00 Hz	: 2007.22
Bits to avoid Startcode Emulation	: 0
Bits for parameter sets	: 176

图 25: 编码结果

对编码视频进行解码重建, HEVC 和 H.264 关键帧的对比如下 (左边为 H.264 的结果, 右边为 HEVC):



图 26: 第 1 帧 (I) 帧对比



图 27: 第 15 帧 (I) 帧对比



图 28: 第 30 帧 (I) 帧对比

而对比比较 HEVC 和 H.264 的文件大小和压缩比，对于同等参数的编码压缩，HEVC 的压缩比更高，二进制码流文件更小。

<input type="checkbox"/> test.264	2019/11/30 17:04	264 文件	246 KB
<input type="checkbox"/> test_dec.yuv	2019/11/30 17:13	YUV 文件	4,455 KB
<input type="checkbox"/> test_rec.yuv	2019/11/30 17:04	YUV 文件	4,455 KB

图 29: H.264 文件大小

rec.yuv	2019/11/30 13:47	YUV 文件	4,455 KB
str	2019/11/30 13:47	BIN 文件	182 KB

图 30: HEVC 文件大小

综上分析，在有 I 帧的时候，H.264 和 HEVC 编解码的效果相当。H.264 速度更快，但是对应的二进制码流文件更大；HEVC 虽然过程较慢，但是压缩比更高。

### 2.3.2 参数二——第一帧 I 帧，其余 P 帧的对比分析

采用“CREW-352x288-15-orig-01.yuv”序列来进行第二个参数的对比在该参数条件下，进行 H.264 的编码过程如下：

0000(NVB)	176								
0000(IDR)	46576	0	28	38.997	43.006	42.127	993	0	FRM 396
0001(P)	10280	0	28	38.373	42.766	41.851	1966	521	FRM 22
0002(P)	17456	0	28	38.217	42.840	41.806	2916	1031	FRM 98
0003(P)	42392	0	28	37.848	40.969	40.440	3911	1569	FRM 274
0004(P)	21688	0	28	38.165	42.465	41.417	4666	2074	FRM 103
0005(P)	21104	0	28	38.340	42.582	41.450	5593	2598	FRM 122
0006(P)	14512	0	28	37.954	42.399	41.233	6480	3146	FRM 25
0007(P)	22912	0	28	37.985	41.545	40.933	7425	3651	FRM 81
0008(P)	19504	0	28	37.996	41.935	40.841	8396	4234	FRM 51
0009(P)	20176	0	28	37.785	42.004	40.853	9291	4707	FRM 66
0010(P)	21720	0	28	37.666	41.778	40.928	10267	5271	FRM 64
0011(P)	20864	0	28	37.645	41.863	40.816	10189	5226	FRM 68
0012(P)	21984	0	28	37.597	41.734	40.757	10256	5260	FRM 61
0013(P)	20960	0	28	37.572	41.683	40.668	10251	5313	FRM 51
0014(P)	24544	0	28	37.544	41.526	40.388	10441	5366	FRM 58
0015(P)	25128	0	28	37.308	41.131	40.196	10267	5237	FRM 41
0016(P)	24272	0	28	37.293	41.334	40.041	10355	5294	FRM 37
0017(P)	26160	0	28	37.306	41.055	39.959	10328	5274	FRM 37
0018(P)	25168	0	28	37.183	41.063	39.917	10245	5242	FRM 43
0019(P)	24192	0	28	37.252	41.229	40.047	10240	5263	FRM 43
0020(P)	24648	0	28	37.263	41.181	39.957	10424	5328	FRM 58
0021(P)	24784	0	28	37.219	41.381	39.979	10339	5288	FRM 56
0022(P)	23248	0	28	37.040	41.065	39.908	10275	5346	FRM 28
0023(P)	26576	0	28	36.945	40.963	39.950	10308	5239	FRM 46
0024(P)	29920	0	28	37.111	40.962	39.940	10255	5228	FRM 74
0025(P)	34680	0	28	37.113	40.932	39.934	10327	5279	FRM 126
0026(P)	31728	0	28	37.061	40.919	39.901	10501	5294	FRM 90
0027(P)	28304	0	28	37.193	40.684	39.874	10456	5292	FRM 66
0028(P)	26152	0	28	37.279	41.119	39.985	10312	5232	FRM 58
0029(P)	25800	0	28	37.317	40.970	40.056	10346	5254	FRM 48

图 31: H.264 编码过程

编码对应的结果如下，其中用时 258.019s，编码 30 帧，PSNR 值为  $SNR_Y = 37.59, SNR_U = 41.57, SNR_V = 40.54$ ：

```

----- Average data all frames -----
SNR Y(dB)      : 37.59
SNR U(dB)      : 41.57
SNR V(dB)      : 40.54
Total bits     : 747608 (I 46576, P 700856, NVB 176)
Bit rate (kbit/s) @ 30.00 Hz : 747.61
Bits to avoid Startcode Emulation : 0
Bits for parameter sets : 176

```

图 32: 编码结果

对编码视频进行解码重建，HEVC 和 H.264 关键帧的对比如下（左边为 H.264 的结果，右边为 HEVC）的结果：



图 33: 第 1 帧 (I) 帧对比



图 34: 第 15 帧 (P) 帧对比





图 35: 第 30 帧 (P) 帧对比

而比较 HEVC 和 H.264 的文件大小, 对于同等参数的编码压缩, HEVC 的压缩比还是更高, 二进制码流文件更小。

test.264	2019/11/30 18:56	264 文件	92 KB
test_dec.yuv	2019/11/30 19:00	YUV 文件	4,455 KB
test_rec.yuv	2019/11/30 18:56	YUV 文件	4,455 KB

图 36: H.264 文件大小

rec.yuv	2019/11/30 16:03	YUV 文件	4,455 KB
str	2019/11/30 16:03	BIN 文件	46 KB

图 37: HEVC 文件大小

综上所述, 在第二种编码方式——第一帧为 I 帧, 其余为 P 帧时。H.264 和 HEVC 编解码效果相当。H.264 速度更快, 压缩比低; HEVC 速度慢但压缩比更高。

### 2.3.3 综合分析对比

序列	编码类型	编码前文件大小	编码后文件的大小	用时
1	HEVC	4455KB	182KB	126.4317s
1	H.264	4455KB	246KB	33.675s
2	HEVC	4455KB	46KB	456.247s
2	H.264	4455KB	92KB	258.019s

表 2: QP 值变化对于同一视频编码结果的影响

综上所述, 根据了两种参数的对比分析。对于已知测试序列, 其实 H.264 和 HEVC 的编码再解码的效果相当。但是 HEVC 的压缩比更高, 这就说明了 HEVC 能够更高效的传输



较大的文件。这也是 HEVC 相比与 H.264 的一个巨大的优势。

### 3 实验感想

通过这一次实验，我对 HEVC 和 H.264 编解码的过程有了更深入的认识。同时也理解到了，HEVC 作为更新一代的技术，他的技术优点。通过动手实践，我也对视频压缩编码的过程和原理有了更多的了解。