AVC 부호화 효율의 추정

융 응옥 튜이 둥, 손원
경희대학교
banhbao1986@khu.ac.kr, wsohn@khu.ac.kr

Estimation of AVC Coding Efficiency

Luong Ngoc Thuy Dung, Won Sohn
Kyung Hee University

Abstract

This study investigates some schemes to estimate the coding efficiency of a video sequence. The texture complexity and motion are considered as two major parameters to decide the coding efficiency, and the methods to estimate the parameters are discussed. For a fixed values of PSNR, the bit rate of a video sequence is estimated using some schemes based on the estimated parameters, and compared with the bit rate by MPEG-4 AVC.

1. Introduction

The coding efficiency of a video sequence is different depending on its contents, and we are trying to classify a video sequence with respect to the coding efficiency. The MPEG-4 AVC is considered as a reference coding scheme.

The texture complexity of a video frame and the level of motion in a video sequence are considered as two factors to affect the coding efficiency of a video sequence. The transform coding gain [], entropy, etc. can be used to represent the texture complexity of a frame. The magnitude of a motion vector is employed to represent the level of motion for a video sequence.

The coding efficiency of a video sequence is measured by estimating a bit rate of a compressed video sequence at PSNR of 30 and 35 dB. The bit rate is estimated by a linear least square estimator based on the average transform coding gain and averaged magnitude of motion vectors for a video sequence.

The three schemes are considered to estimate the bit rate. We applied the linear least square estimator to estimate the bit rate. The first scheme is to estimate a parameter set for all the video sequences, but we have large errors. The second one is to estimate a parameter set for each video sequence, but it is not realizable in practice. The last one is to estimate one parameter set for the whole video sequences, but it is a function of texture complexity and motion parameters for each video sequence.

This paper is organized as follows. Chapter 2 illustrates problem formulation and coding efficiency of AVC. Chapter 3 presents measuring schemes for classification of video sequences. Chapter 4 proposes three coding efficiency estimation methods. Chapter 5 concludes our work.

2. Problem formulation

The coding efficiency of AVC depends on different video sequences. We are trying to propose AVC coding efficiency estimation schemes for any video sequence without compressing the sequence.

Video sequences analysis consists of motion parameter and texture complexity analysis. Motion characteristics for a video sequence can be estimated by using full search algorithm [4]. The texture complexity parameter can be obtained by computing transform coding gain for video sequences. The study will propose coding efficiency estimation methods through motion and texture complexity parameters for video sequences.

To analyze the performance of AVC, we used the JSVM software version 9.19.7 [3]. Table I shows the coding efficiency of AVC for nine typical video sequences when PSNR = 30, 35 dB.

Table I. Coding efficiency of AVC

Video sequence	Motion characteristic	Texture complexity	bit rate (kbps)	
			30 dB	35 dB
Bus	Slow	Low	78	135
City	Slow	Medium	80	190
Mobile	Slow	High	377	1300
Crew	Medium	Low	126	442
Foreman	Medium	Medium	91	227
Harbour	Medium	High	361	1205
Football	Fast	Low	128	357
Ice	Fast	Medium	223	810
Soccer	Fast	High	267	846

Table 1 shows the classification of each video sequence with respect to motion and texture complexity, and bit rates by AVC when PSNR=30 and 35 dB. It can be seen that the bit rates are changing depending on a video sequence. Therefore, we try to find how to estimate coding efficiency using characteristics of each video sequence.

3. Measuring Schemes

This section will present the measuring parameters for video sequences. Video sequence analysis consists of motion parameter and texture complexity analysis. Thus, we could classify video sequences based on analyzing the parameters.

3.1 Motion parameter

Motion parameter is the averaged magnitude of motion vectors for video sequence. We can use the Full Search algorithm to find the motion vectors for each frame or a frame for GOP. The algorithm calculates the cost function at each possible location in the search window. As a result of which it finds the best possible match and gives the least Mean Absolute Difference (MAD) or the least Sum Absolute Difference (SAD) defined in eqs. of (1) and (2)

$$MAD = \frac{1}{N^2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\left|C_{ij} - R_{ij}\right| \tag{1}$$

$$SAD = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\left|C_{ij} - R_{ij}\right| \tag{2}$$

where N×N is the size of a macroblock, C_{ij} and R_{ij} are intensities of pixel being compared in current macro block and reference macro block.

We will find row and column ordinate (M_x and M_y) for motion vector of macro block which has minimum MAD/SAD. The magnitude of motion vector for macro block can be defined in (3)

$$M = \sqrt{M_x^2 + M_y^2} \tag{3}$$

We know that the most common pixel format is the byte image, where this number is stored as an 8-bit integer giving a range of possible values from 0 (black) to 255 (white). After analyzing nine sequences, we also see that the average SAD of no matching block is within the range of 500~1300. Thus, we propose the threshold M to detect no block matching case. If SAD of macro block is less than M, we can find out the no matching block. We can choose M=550 or 600. Higher threshold will result larger detection error.

3.2 Texture complexity parameter

We can classify the texture complexity for video sequence through transform coding gain for n^{th} frame is defined in (4)

$$G_{TC}(n) = \frac{\frac{1}{N}\sum_{i=0}^{N-1}\sigma_i^2(n)}{\left(\prod_{i=0}^{N-1}\sigma_i^2(n)\right)^{\frac{1}{N}}} \tag{4}$$

where σ_i^2 is the variance of the i^{th} DCT coefficient for a macro block, n is the number of frames for test sequence, and N is the number of MB for a frame. If we use CIF sequence (352x288) and 16x16 Macro block size, N will equal 396 MB. $\left(N = \frac{352\times 288}{16^2} = 396\right)$

We classified the motion parameter and texture complexity parameters as follows:

(1) Averaged magnitude of motion vector:
0~2: Slow, 2~5: Medium, 5~: High
(2) Averaged DCT transform coding gain:
0~30: Low, 30~65: Medium, Above 65: High

Table II shows transform coding gain and magnitude of motion vectors results for all test sequences.

Table II. Transform coding gain and magnitude of motion vector

Test sequence	Motion characteristic	Texture complexity	Average transform coding gain	Average magnitude of motion vector
Bus	Slow	Low	15.6	0.3
City	Slow	Medium	46.9	1.8
Mobile	Slow	High	82.7	1.0
Crew	Medium	Low	29.8	4.7
Foreman	Medium	Medium	52.3	3.4
Harbour	Medium	High	88.7	4.8
Football	Fast	Low	13.3	8.2
Ice	Fast	Medium	42.3	5.3
Soccer	Fast	High	130.5	6.3

Through table II, we can see that video sequences which have the highest motion will have the largest averaged magnitude of motion vector such as Ice, Football and Soccer. It is similar for texture complexity case. Video sequences which have the least texture complexity will have the smallest averaged transform coding gain.

4. Estimation of coding efficiency

This section propose three coding efficiency estimation schemes through motion and texture complexity parameters by using multiple linear regressions. The estimated bit rate is defined as follows.

$$\hat{r} = \alpha\cdot m + \beta\cdot tc + \gamma \tag{5}$$

Here, $\hat{r}$ is an estimated bitrate, m is an averaged magnitude of motion vectors, and tc is an averaged transform coding efficiency.

4.1 Method 1

This method will estimate coding efficiency through motion and texture complexity parameters for all video sequences to find a set of α, β, γ coefficients.

$$\hat{r}_i = \alpha\times m_i + \beta\times tc_i + \gamma \tag{6}$$

where $\hat{r}_i$ is the estimated bit rate for video sequence.

m_i, tc_i are motion and texture complexity parameters for each video sequence. Estimation error is defined as follows.

$$\text{error}=\left|\frac{\text{estimated bitrate - measured bitrate}}{\text{measured bitrate}}\right| ?100\ (\%) \tag{7}$$

Here the measured bit rate is the bit rate measured by JSVM. Table III-IV shows the bit rate estimation through motion/texture parameters at PSNR=30/35dB.

Table III. Estimation coding efficiency when PSNR=30dB

Video sequence	α=8, β=287, γ=66		
	Measured bit rate	Estimated bit rate	Error (%)
Bus	78	100.7	29
City	80	136.4	70
Mobile	377	248.0	34
Crew	126	171.1	36
Foreman	91	184.1	102
Harbour	361	266.0	26
Football	128	102.3	20
Ice	223	162.9	27
Soccer	267	359.5	35

Table IV. Estimation coding efficiency when PSNR=35dB

Video sequence	α=64, β=1018, γ=145		
	Measured bit rate	Estimated bit rate	Error (%)
Bus	135	270.2	100
City	190	416.4	119
Mobile	1300	794.9	39
Crew	442	526.3	19
Foreman	227	579.3	155
Harbour	1205	876.2	27
Football	357	311.7	13
Ice	810	513.3	37
Soccer	846	1213.8	43

The advantage of method 1 is simple and easy to estimate coding efficiency. However, the error is too big. Thus, it is not effective method. We need to find another method to decrease these big errors.

4.2 Method 2

In this method, we also estimate coding efficiency through motion and texture complexity parameters for video sequences. It is nearly similar to method 1. However, bit rate, motion and texture parameters are applied for each GOP (Group Of Pictures). It means that each video sequence will have a corresponding set of α, β, γ coefficients.

$$\hat{r}_i = \alpha \times m_i + \beta \times tc_i + \gamma \qquad (8)$$

where $\hat{r}_i$ is the bit rate for each GOP, and m_i, tc_i are motion and texture complexity parameters for each GOP.

The α, β, γ coefficients are shown in Table V and VI

Table V. Estimated coefficients α, β, γ (PSNR=30 dB).

Sequence	α	β	γ
Bus	0.0575	-0.0294	11.7943
City	0.0442	-0.0160	8.5781
Mobile	0.0526	-0.0002	0.4639
Crew	0.0152	-0.0233	16.6846
Foreman	0.0281	-0.0134	9.4432
Harbour	0.0153	0.0025	4.3434
Football	-0.0185	-0.0325	23.5445
Ice	0.0101	-0.0184	14.2162
Soccer	0.0001	0.0190	-1.2721

Table VI Estimated coefficients α, β, γ (PSNR=35 dB).

Sequence	α	β	γ
Bus	0.1019	-0.0531	23.2751
City	0.0756	-0.0305	15.1755
Mobile	0.0843	0.0018	-0.0494
Crew	0.0292	-0.0493	32.5503
Foreman	0.0483	-0.0251	17.1469
Harbour	0.0247	0.0072	7.2284
Football	-0.0324	-0.0568	43.0751
Ice	0.0172	-0.0328	29.1955
Soccer	-0.0001	0.0434	-4.6351

Table VII- VIII shows coding efficiency estimation and the error between measured and estimated bit rate by using method 2 when PSNR = 30, 35dB.

Table VII. Estimation of coding efficiency (PSNR=30dB).

Video sequence	Measured bit rate	Estimated bit rate	Error (%)
Bus	78	83	6
City	80	89	11
Mobile	377	414	10
Crew	126	141	12
Foreman	91	95	5
Harbour	361	387	7
Football	128	146	14
Ice	223	246	10
Soccer	267	295	11

Table VIII. Estimation of coding efficiency (PSNR=35dB).

Video sequence	Measured bit rate	Estimated bit rate	Error (%)
Bus	135	142	5
City	190	210	11
Mobile	1300	1384	6
Crew	442	490	11
Foreman	227	241	6
Harbour	1205	1264	5
Football	357	403	13
Ice	810	892	10
Soccer	846	936	11

We can see that through table VII~VIII, the error between measured and estimated bit rate is much less than method 1. However, through table V and VI, it can be seen that α, β, γ coefficients depend on video sequence. It means that these coefficients are different when motion and texture complexity parameters are different. By this method, we can not specify which α, β, γ coefficients are suitable for any video sequence because through method 2 we find out the different α, β, γ coefficients for each video sequence. Thus, it is necessary to find the method which is general for all cases with low error between original and estimated bit rate.

4.3 Method 3

In this method, we will estimate α, β, γ coefficients through other parameters. Specifically, α is estimated through motion parameter, β is estimated through texture complexity parameter and γ is estimated through both motion and texture complexity parameter.

$$\begin{aligned} \alpha_i &= a \times m_i + b \\ \beta_i &= c \times tc_i + d \\ \gamma_i &= e \times m_i + f \times tc_i + g \end{aligned} \qquad (9)$$

where $\alpha_i, \beta_i, \gamma_i$ are coefficients for each video sequence.

m_i, tc_i are the averaged magnitude motion vector and DCT transform coding gain for video sequence.

Through a, b, c, d, e, f, g coefficients, we can estimate bit rate for AVC bit stream with low error when comparing with original bit rate.

Table IX. Coefficients a, b, c, d, e, f, g.

	30 dB	35 dB
a	0.0004	0.0008
b	-0.0333	-0.0644
c	-0.0091	-0.0158
d	0.0574	0.0994
e	1.2791	2.4930
f	-0.1768	-0.3631
g	13.7472	27.5709

Table X. Estimation of coding efficiency (PSNR=30dB).

Video sequence	Measured bit rate	Estimated bit rate	Error (%)
Bus	78	85	9
City	80	98	23
Mobile	377	476	26
Crew	126	162	29
Foreman	91	101	11
Harbour	361	402	11
Football	128	157	23
Ice	223	268	20
Soccer	267	297	11

Table XI. Estimation of coding efficiency (PSNR=35dB).

Video sequence	Measured bit rate	Estimated bit rate	Error (%)
Bus	135	146	8
City	190	212	12
Mobile	1300	1395	7
Crew	442	567	28
Foreman	227	245	8
Harbour	1205	1324	10
Football	357	409	15
Ice	810	1047	29
Soccer	846	1182	40

Nine sequences fully represent for different video sequences through both motion and texture complexity parameters. Through table X-XI, it can be seen that we can estimate bit rate of AVC using motion and texture complexity parameters. Thus, with any video sequence, by analyzing its characteristic, we can classify through motion and texture parameter as well as estimate the coding efficiency.

4. Conclusion

The AVC coding efficiency depends on video sequences. It means that different video sequence will have different coding efficiency. Besides, video sequence can be classified through motion and texture complexity parameters. Motion parameter is the averaged magnitude of motion vector and texture complexity is the averaged DCT transform coding gain for video sequence. Thus, we have proposed AVC coding efficiency estimation methods through motion and texture complexity parameters with the low error between measured and estimated bit rate.

References

[1] ISO/IEC 14496-10 and ITU-T Rec.H.264: "Advanced Video Coding", 2003.
[2] Mathias Wien, H.Schwarz, Tobias Oelbaum, "Performance Analysis of SVC", IEEE Transactions On Circuit and Systems For Video Teachnology, Vol.17, No.9, September 2007.
[3] JSVM software, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6)
[4] Yeu-Meng Chen, Ivan V.Baji, "Motion Vector Outlier Rejection Cascade for Global Motion Estimatation", IEEE Signal Processing letters, Vol. 17, NO. 2, February 2010.
[5] Wonsang You, M.S. Houari Sabirin, Munchurl Kim, "Moving Object Tracking in H.264/AVC Bitstream", Springer-Verlag Berlin Heidelberg 2007.
[6] Khalid Sayood, "Introduction to Data Compression", Second edition, ISBN 1-55860-558-4.