Adaptive I frame Positioning For MPEG Bit Rate Improvement

H. Farouk
Computers and Systems Dept.
Electronic Research Institute
El-tahrir st., Dokki, Cairo,
EGYPT
http://www.eri.sci.eg

Abstract:- MPEG coding algorithm is a full motion compensated DCT. In MPEG coding, the video sequence first divided into groups of picture of frames (GOP), (I,P,B frames). The algorithm for adaptive I frame position, depending on the indexing technique, In This paper an algorithm was developed to variable GOP length which was resulted in minimizing the bit rate with average 10% to 15% from the classical MPEG coding technique while keeping the same SNR with respect to the standard MPEG coding algorithm.

Keyword: video compression, motion compensation, GOP, bit rate, video indexing, shot cut

1 Introduction

MPEG coding algorithm is a full motion compensated DCT. In MPEG coding, the video sequence first divided into groups of picture of frames (GOP). Each group may include three types of pictures or frames: intra code of (I) picture of frame, Predictive – coded (P) picture of frame and bidirectional productively coded (B) frame or picture. I pictures are coded by intra frame technique only with no need for previous information. In other words, I pictures are self-sufficient. They are used as anchors for forward and/or backward prediction. P-pictures are coded using one-directional motion compensated prediction from a previous anchor frame, which could be either I- or P-picture. The distance between two nearest I-frames is denoted by *N*, which is the size of GOP. The distance between two anchor frames is denoted by *M*. Parameters *N* and *M* both are user selectable parameters, which are selected by the user during the encoding. A large number of *N* and *M* will increase the coding performance but cause error propagation or drift. Usually, *N* is chosen from 12 to 15 and *M* from 1 to 3. If *M* is selected to be 1, this means no B-picture will be used. Last, P picture can be coded using prediction form either past or future anchor frames (I or P), or both. [1-5]

Section 2 will discuss the related work. Section 3 will discuss why we need adaptive I frame and then introduce the way to make I frame adaptive, using video indexing and shot cut detection techniques, in section 4. The results will be given in section 5. The conclusion will be in section 6

2 Related Work

The best work in this area was done by presenting a new algorithm that adaptively selects the best possible reference frame for the predictive coding of generalized, or multi-view, video signals, based on estimated prediction similarity with the desired frame. They defined similarity between two frames as the absence of occlusion, and they estimated this quantity from the variance of composite displacement vector maps. The composite maps are obtained without requiring the computationally intensive process of motion estimation for each candidate reference frame. They provided prediction and compression performance results for generalized video signals using both this scheme and schemes where the reference frames were heuristically pre-selected. When the predicted frames were used in a modified MPEG encoder simulation, the signal compressed using the adaptively selected reference frames required, on average, more than 10% fewer bits to encode than the non-adaptive techniques. The disadvantage of this technique is the computational complexity in calculating the similarity between the frames [6].

3 Why to Make I Frame Adaptive?

The value of *N* and *M* is fixed for a session i.e. for one encoding session. But in our work we can change the value of *N* to fit the shot cut since we will program the encoder to access data file with the frames number represent shot start which is the output of the indexing algorithm described in [7]. The fixing of *N* and *M* ignore the nature of shots with respect to its content by meaning that sometimes we have motion with short shots or fast motion, fast content changes internally on the frame level, or long shots with slow changes and so on.

In this work we take into consideration the nature of video content with respect to the speed of changes and the shot cut to change *N* within the same session. With variable *N* we can select the location to use I frame, this location will help in optimizing the bit rate since we can put I frame as far as possible during slow motion or long shots. Because with the classical MPEG coding algorithm we can consider I frame within the same shot, where it is preferred to code all relative frames as P or B frame as long as we are in the same shot. Coding Equal distance I frame, as classical MPEG does, will produce a large bit rate as table 1 tell us that the average bit rate for I frame (150 kbit/sec) is 3 times the average bit rate of frame P (50 kbit/sec) or 7.5 times the average bit rate of B frame (20 kbit/sec).[8]

Level	I	P	B
30 Hz SIF @ 1.15 Mbit/sec	150,000	50,000	20,000
30 Hz CCIR 601 @ 4 Mbit/sec	400,000	200,000	80,000
Level	**Average**		
30 Hz SIF @ 1.15 Mbit/sec	38,000		
30 Hz CCIR 601 @ 4 Mbit/sec	130,000		

Table 1: Pictures average bit rate

4 How To Make I Frame Adaptive?

The I frame is made adaptive with variable position in the coding sequence since each frame type will be marked with its type through the coded video stream. We can use one or more of the following techniques:

1. Use the output of the algorithm described in [7] for indexing video and detect the shot cut position which will be based on implementing a technique that depends on histogram differences of the DC coefficients of the DCT blocks of coefficients in the compressed domain, or just compute the DC coefficient in the spatial domain. There is no need to work on the whole coefficients, The DC coefficient only is effective because it contain the most entropy of the spatial block data either it is shot cut or gradual effect. We will consider allocating of shot cut is the allocation of frames exceeds certain threshold as shown in figure 2. The position of shot cut will be consider the position of new I frame in the position of shot cut is mandatory as starting new frame information. This will help in bit rate minimization. We use this method in our technique.

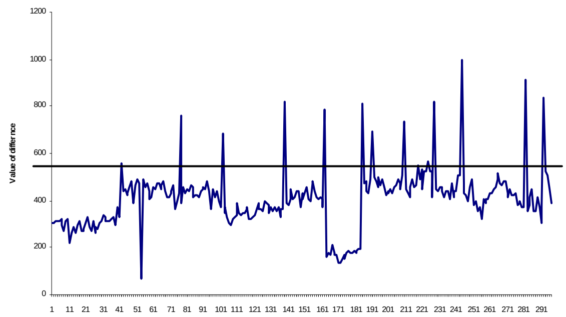

Figure 1: Histogram differences with threshold for shot cut detection

2. If the video shot (after indexing) is *pan* or *tilt* or *zoom* we can determine a relation between the numbers of pixels shift from one frame to the next one and if there is a need for a new frame and depending on motion compensation algorithm will be sufficient in coding these frames within the same shot (same camera mtion directions) [6]. This will help in both encoding time and bit rate minimization as it could be applied online [9-11].

5 Results

The tests given below keep the same SNR, as shown in Figure 1, because this algorithm may searches for a not found block and will code it as a new one.

In our tests we apply many cases. We consider: GOP=40, GOP=60, GOP=240, GOP=300 and finally put GOP >= Frames number of the tested sequence. The tests given below are for GOP>= Frames number of the tested sequence

Test 1:

We apply test case depending on the indexing algorithm for a sequence of 300 frames (the test sequence of the indexing algorithm which is containing 12 keyframes) by taking F=300 and consider the keyframes resulting from the indexing algorithm. This gives a SNR difference around zero between the coding with variable I frame position and the classical MPEG coding algorithm as shown in Figure 2.

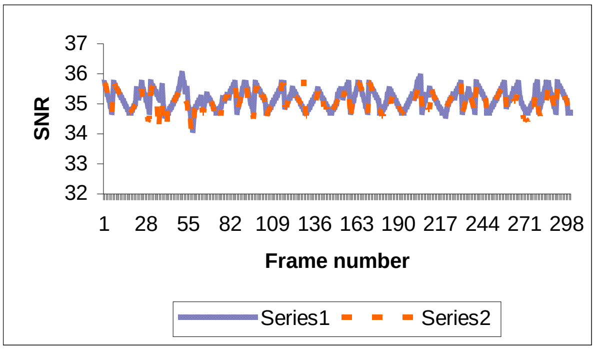

Figure 2: SNR for sequence Nike1.mpg coded with MPEG with respect to the original stream (series 1) is identical to sequence processed with adaptive I frame with respect to the original stream (series2).

Consider MPEG stream with GOP=15

$\Rightarrow$ IBBPBBPBBPBBPBP

with N=15, M=3 Where:

N is the length of GOP

M is the distance between I and P

With classical MPEG coding, the 300 frames will be divided into:

Number of GOP = 300 /15 =20

I frames = 1x20 = 20 (1 per GOP)

P frames =5x20 = 100 (5 per GOP)

B frames = 9x20 = 180 (9 per GOP)

According to table 1: this video sequence average bit rate is

= (20x150) + (100x50) + (180x20)/ (no. of sec.)

= 11600/10 kbits/sec

= 1160 kbits/sec

With adaptive I frame, we will consider the I frame position is only the position of keyframes resulted from the indexing technique described in [6-7]

So, the 300 frames are divided into:

1 I frames = 12 frames
(The number of keyframes)

2 P frames = (300 –12)/3 =96 frames

(Dividing over 3 because the predicted frames P and the bi-directional prediction B will compose groups of (BBP) so, P will be repeated each 3 frames)

1 B frames = 300 – (12 + 96)= 192 frames
According to table 1 : this video sequence bit rate average is

= (12x150)+(96x50)+(192x20)/no. of seconds

= 10440/10 kbits/sec

= 1044 kbits/sec

The bit rate is minimized by the factor:

(1-(1044/1160))x100 = 10 %

Test 2:

In this test we use a part of an actual movie (home alone 3):

From frame 7:9:00 to Frame 11:19:00 with frame rate 30 frame/sec.

(The format is Minutes:Second:frame number)

The number of frames in this test

= ((11-7) x 60 + (19-9)) x 30

= 7500 frames

The number of keyframes in this test = 60 frames

For the same GOP as the previous test:

With classical MPEG coding, the 7500 frames are divided into:

Number of GOP = 7500 /15 =500

I frames = 1x500 = 500 (1 per GOP)

P frames = 5x500 = 2500 (5 per GOP)

B frames = 9x500 = 4500 (9 per GOP)

According to table 1 : this video sequence bit rate average with **classical MPEG coding is**:

= (500x150) + (2500x50) + (4500x20)/ (no. of sec.)

= 290000/250 kbits/sec

= 1160 kbits/sec

I frames = 60 frames

P frames = (7500 – 60)/3 = 2480 frames

B frames = 7500 – (60 + 2480)

= 4960 frames

According to table 1 and by adding 10% of the average P type size to P type itself as the not found block will be coded as still block. This is not the same for test 1. Test 1 is composed of rather small shots. So, this video sequence average with

adaptive I frame for test 2 is
= (60x150) + (2480x55) + (4960x20)/ (no. of sec.)
= 244600/250 kbits/sec
= 978.4 kbits/sec.

The bit rate is minimized by the factor:
(1-(978.4/1160)) x100 = 15.6 %

So, from this tests, test1 and test2:
Test 1 represents short shots with many I frames required and the bit rate for frames number 300. The minimization average is 10%
Test 2 represents real movie with smooth, moderate width shot with relatively smaller number of I frames required for frames number 7500. The minimization average is 15%.

6 Conclusion

The algorithm for adaptive I frame position, depending on the indexing technique, results in minimizing the bit rate with average 10% to 15% from the classical MPEG encoding technique while keeping the same SNR with respect to the standard MPEG encoding algorithm. The overhead is the video indexing and shot cut detection calculations or any algorithm for keyframe extraction. But this will not consider a problem because the adaptive I frame position technique depends on historical calculations (offline processing) on the whole set of frame composes the sequence gave an indexed video output which is another useful application. This means that we have no private overhead we consider this technique as an application to the indexing. This relies that this work will help in bit rate improvement for the offline applications and will help developing techniques for intelligent retrieval from digital video libraries.

7 References

[1] ISO/IEC 11172: 'Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s'.

[2] ISO/IEC 13818: 'Generic coding of moving pictures and associated audio (MPEG-2)'.

[3] Online web site: http://www.mpeg.org.

[4] Yun Q. Shi Huifang Sun, “Image and Video Compression for Multimedia Engineering”, CRC press,1999.

[5] C. Stiller and J. Konrad, “Estimating motion in imaging sequence” IEEE Signal Processing Magazine, pp. 70-91, July, 1999.

[6] S. L. Regunathan, R. Zhang and K. Rose, "Scalable video coding with robust mode selection," Signal Processing: Image Communication, vol.16, (no.8), Elsevier,. p.725-32, May 2001

[7] H. Farouk, S. Mashali, M. Rashwan, A. Nassar, "Video Compression Improvements Based On Camera Motion" , IEEE ISSPIT Dec. 28-30, 2001.

[8] R. Zhang, S. Regunathan and K. Rose, "Optimized Video Streaming Over Lossy Networks With Real-Time Estimation Of End-To-End Distortion," International Conference on Multimedia and Expo 2002, Lausanne, Switzerland, August 2002.

[9] R. Brunelli, O. Mich, and C. M. Modena, “A survey on the automatic indexing of video data”, Journal of visual communication and image representation, pp.78-112, 1999.

[10] Zhang, H.J., Kankanhalli, A., and Smoliar, S.W., "Automatic Partitioning of Full-motion Video", Multimedia Systems (1993) Vol. 1, No. 1, pp. 10-28.

[11] W. Wolf, "Keyframe Selection By Motion Analysis", in Proc. IEEE, Conf. Acoust., Speech, and Signal Proc., 1996.