Nigerian Journal of Technology (NIJOTECH)
Vol. 32. No. 3. November 2013, pp. **522 – 527**
 ISSN 1115-8443
www.nijotech.com

LOW BIT RATE VIDEO CODING

S. Mishra[1], S. Sawarkar[2]
[1]DEPARTMENT OF EXTC, THAKUR COLLEGE OF ENGINEERING AND TECHNOLOGY, UNIVERSITY OF MUMBAI, INDIA
[2] DATTA MEGHE COLLEGE OF ENGINEERING, UNIVERSITY OF MUMBAI, INDIA
E-mail addresses: [1] sangeeta.mishra@thakureducation.org, [2]sudhir_sawarkar@yahoo.com

Abstract
Variable length bit rate (VLBR) broadly encompasses video coding which mandates a temporal frequency of 10 frames per second (fps) or less. Object-based video coding represents a very promising option for VLBR coding, though the problems of object identification and segmentation need to be addressed by further research. Pattern-based coding is a simplified object segmentation process that is computationally much less expensive, though a real-time, content-dependent pattern generation approach will certainly improve its acceptance for VLBR coding. In this paper pattern based coding technique is used. In this paper, a very low bit-rate video coding algorithm that focuses on moving regions is performed. The aim is to improve the coding performance, which gives better subjective and objective quality than that of the conventional coding methods at the same bit rate. Eight patterns are pre-defined to approximate the moving regions in a macroblock. The patterns are then used for motion estimation and compensation to reduce the prediction errors. Furthermore, in order to increase the compression performance, the residual errors of a macroblock are rearranged into a block with no significant increase of high-order DCT coefficients. As a result, both the prediction efficiency and the compression efficiency are improved. This paper shows that using pattern based coding the compression ratio is better.

Keywords: Motion estimation, video compression, MPEG, H.264, EZW.

1. Introduction

Coding for low bit rate video applications has gained a special interest among the video coding community especially with the emergence of many applications such as video conferencing, video telephony, surveillance, and monitoring. In each case, video and audio information are transmitted over telecommunication links, including networks, telephone lines, ISDN and radio. The bandwidth required for the transmission of digital video is very much insufficient. So the compression of digital video is required to reduce the rate of information that is to be transmitted using the telecommunication links.

The integration of motion video as an integral part of multimedia environments is technologically one of the most demanding tasks, due to the high data rates and real-time constraints. According to the importance of the problem, there is a significant amount of ongoing research in the area of video compression.

Demand of video applications such as video telephony, video conferencing, online stream video, mobile streaming, Television (TV), 24 hours surveillance system and many others are increasing exponentially because of the evolution of video compression technology. Bandwidth and storage space are crucial parameters in video communication. Their requirements are also increased exponentially with increasing demand of video applications. New video compression standard needed to satisfy growing demand of video applications and be more effective than previously developed standards. It should use less bandwidth, and storage space as compared to previous video codec standards [1].

Two major groups are actively participating in enhancement of video codec, first is Video Coding Experts Group (VCEG) from International Telecommunications Union(ITU), and the other is Moving Picture Experts Group (MPEG) from International Organization for Standardization (ISO)/International Electro-technical Commission (IEC). The first video codec standard is H.120 developed in 1984 by ITU-T. It is lossy. The newer standard defined in 1990 by ITU-T, is called H.261 [7-9]. It is popular for video conferencing and telephony. ISO/IEC also released MPEG-1 part 2 in 1993, which is implemented in video CD [2]. In 1995 ITU-T and ISO/IEC jointly released their new video codec known as H.262, popular in Video CD/DVD/Blue-ray, Video broadcasting. In 1995, H.263 and 1999 MPEG-4 part 2 are developed by ITU-T and ISO/IEC respectively. Finally, H.264 video

2[3-4] codec was announced by ITU-T and ISO/IEC, which is popular for all applications mentioned above [6-10].

2. Video compression

With the advent of the multimedia age and the spread of Internet, video storage on CD/DVD and streaming video has been gaining a lot of popularity. The ISO Moving Picture Experts Group (MPEG) video coding standards pertain towards compressed video storage on physical media like CD/DVD, where as the ITU addresses real-time point-to-point or multi-point communications over a network [2]. The former has the advantage of having higher bandwidth for data transmission. In either standard the basic flow of the entire compression decompression process is largely the same and is depicted in Fig.1 shows the block diagram for video compression process. The most computationally expensive part in the compression process is the Motion Estimation. Motion Estimation examines the movement of objects in sequence to try to obtain the vectors representing the estimated motion. The encoder estimates the motion of the current frame with respect to previous frame. A motion compensated image of the current frame is then created. Motion vector is then transmitted to decoder. The decoder reverses the whole process and creates a full frame. In this way motion compensation uses the knowledge of object motion to achieve data compression.

The motion vectors for blocks used for motion estimation are transmitted, as well as the difference of the compensated image with the current frame is also encoded and sent. The encoded image sent is then decoded at the encoder and used as a reference frame for the subsequent frames. The decoder reverses the process and creates a full frame.

3. Low bit-rate video coding using fixed patterns

The algorithm proposed by Wong et al [12], used eight fixed patterns (Fixed 8), with macroblocks classified according to the following three mutually exclusive classes: 1) Static macroblock (SMB): Blocks containing little or no motion; 2) Active macroblock (AMB): Blocks that contain moving object(s) with little static background; and 3) Active-Region macroblock (RMB): Blocks that contain both static background and some part(s) of moving object(s) [14]. In [11], a pattern codebook of four 128-pixel patterns was used. Further improvements were obtained in the Fixed 8 algorithm by using a pattern codebook of eight 64-pixel patterns (P_1.P_8 in Figure 2). Each pattern consists of total 256 (16×16) pixels. Shaded region in the pattern represents 1's and non-shaded region in the pattern represents 0's. The total number of 1's in the pattern is 64 and hence it is called 64-pixel pattern.

3.1 Moving Region Detection

The basis of this technique is to let the first eight patterns P_1–P_8 in Figure 2 approximate the moving region. Let $C_k(x,y)$ and $R_k(x,y)$, $0 \le x,y \le 15$ denote the k^{th} block of the current and the reference frames respectively. The moving region in the k^{th} block of the current frame is obtained as follows:

$$M_k(x,y)=T(|C_k(x,y)*B-R_k(x,y)*B|) \quad (1)$$

where B, a square pattern of size 3×3, is the structuring element of morphological closing operations, |v| returns the absolute value of v, T(v) returns 1 if v > 2 or 0 otherwise, and , $0 \le x,y \le 15$.

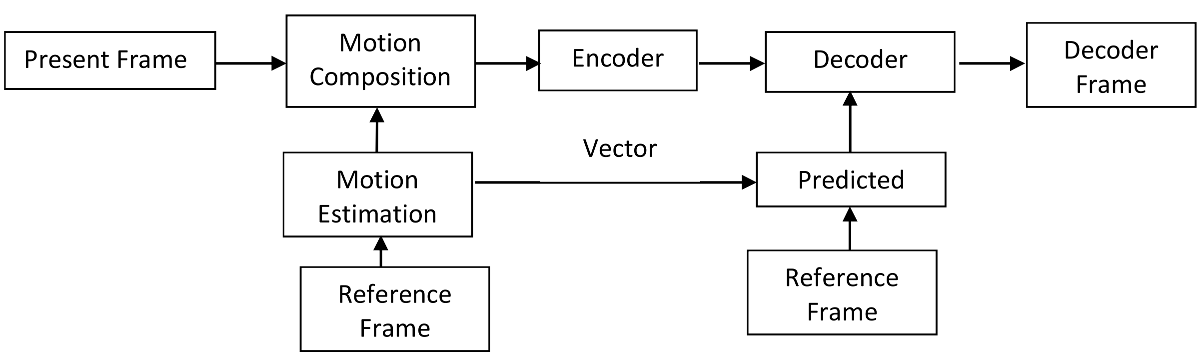

Fig.1: Block Diagram for Video Compression Process flow.

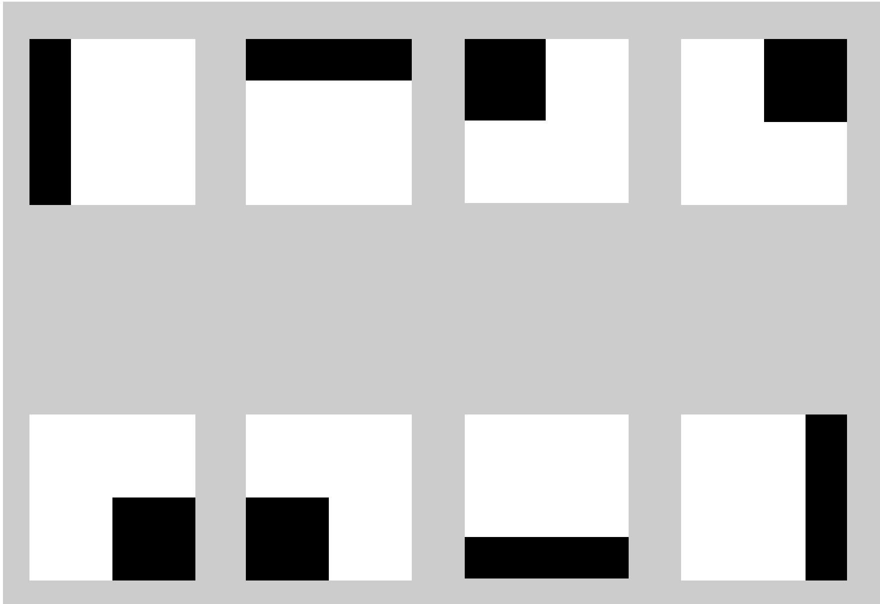

Figure 2: Eight regular shaped 64-pixel patterns, defined in 16×16 blocks, where the shaded region represents 1's and the white region represents 0's.

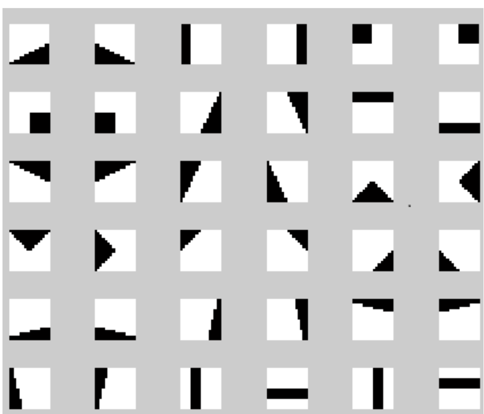

Figure 3: Thirty-six regular shaped 64-pixel patterns, defined in 16×16 blocks, where the shaded region represents 1's and the white region represents 0's.

Each block is then classified into SMB, AMB, and RMB according to the following rules. For the k-th block, if it has less than eight 1's then the block is classified as an SMB; else the block is divided into four sub-blocks and if none of these sub-blocks contain all 0's, the block is classified as AMB. Otherwise, the block is considered as a candidate RMB [12]. Each of these candidate RMBs is then matched against all thirty two prescribed patterns and the best-match pattern is obtained by minimizing the following expression:

$$Dk,n = 1/256(\sum_{x=0}^{15}\sum_{y=0}^{15}|Mk(x,y) - Pn(x,y)|) \quad (2)$$

where 1≤x,y≤36. A candidate RMB is classified as an RMB if min(D_k(x,y)) <0.25; otherwise it is an AMB.

3.2 Motion Estimation and Compensation

Since both SMB and the static regions of RMBs are considered as having no motion, they can be skipped from coding and transmission as they can be obtained from the reference frame. For each AMB, as well as the moving region of each RMB, motion vector and residual errors are calculated using conventional block-based methods, with the obvious difference in having the shape of the blocks for the moving regions of RMBs as that of the best-match pattern, rather than being square [13].

4. Variable Block Motion Compensation

The variable block motion compensation (VBMC) merges normal size of a macro-block into a block of varying size, and the VBMC considers the motion in each macro-block whether to merge them into a larger block. The fixed block motion compensation (FBMC) has many disadvantages. The main disadvantages are that the encoded frame loses a motion inside of the macro-block due to fixed block size, and all motion vectors have to send over the network even though most of information is same in a frame. Those disadvantages can be improved by VBMC technique. The VBMC has block of varying size, so it can capture a small motion by dividing a block into a sub-blocks. The VBMC can avoid the transmission of all motion vectors by merging blocks into a larger block, and send only single motion vector of large block. The proposed motion compensation methodology is based on a bottom-up merging process. It is implemented by merging macro-blocks into large rectangular region under preset threshold value. The proposed technique utilizes the fact that motion vectors of neighboring macro-blocks have high correlation among them. In most of a video data, the background either moves slowly or stands still in the video pictures. The motion vectors representing the still background; can be merged before transmission. The main idea is to combine macro-blocks of still background into a single macro-block of still background. The storage space, bit-rate and bandwidth utilization can be reduced significantly by applying proposed bottom-up merging algorithm. The proposed scheme is very effective in moving picture, whose background and objects are moving in the same direction. The algorithmic details are discussed below.

The first step of motion compensation based on the bottom-up merging procedure is to find out motion vector of current frame by any kind of motion estimation algorithm. Once, the motion vectors are available to motion compensation module, then the bottom-up merging process is implemented in two steps. Firstly, the VBMC merges macro-block into the bigger block, and secondly, it selects one of the merged rectangular blocks among multiple rectangular regions [14]. The VBMC is encoding a frame in raster scan order. It starts merging at each macro-block. The decision of merging should be taken after evaluating a difference of root macro-block and next macro-blocks in raster scan in line. If the difference of two macro-blocks is less than the adjustable threshold, the two macro-blocks are eligible to be merged into a region. The merged macro-block is known as the matched macro-block. The first block is known as root block, because it starts merging process. This evolution process continues until the first mismatch occurs in the raster scan order.

5. Algorithm

The following steps discuss about the mentioned method's implementation

1. Read the video.
2. Divide the video into frames and convert each frame into image.
3. Apply morphology operation using equation number 1 and store it in matrix M.
4. Divide the matrix M into blocks of 16×16.
5. Create the 36 patterns, each of size 16×16 and store it in matrix P_1 ,P_2,P_3 and so on till 36.
6. Calculate matrix D using the matrix M and the pattern matrices P_1, P_2, P_3 etc.
7. Using matrix D calculate RMB if min $(D_k(x,y)) < 0.25)$ else it is an AMB and when there is no motion it is SMB.

6. Results and Discussion

Various videos were observed for encoding and decoding purpose. Since each SMB and the static regions of RMBs are considered as having no motion, they can be omitted from both coding and transmission as they can be obtained from the reference frame. For each AMB, as well as the moving region of each RMB, motion vectors and residual errors are calculated using conventional block-based methods, with the obvious difference in having the shape of the blocks for the moving regions of RMBs as that of the best-match pattern, rather than being square. The number of pixels original defines the total number to represent the particular video. The number of pixels encoded defines the total pixels to represents the encoded video.

Table 1. Comparison of different videos for various parameters

Video	Size	No. of pixels Original	No. of pixels Encoded	No. of AMB	No. of RMB	No. of SMB	Compression Ratio (%)
Rhinos	25MB	26265600	19046400	1	11998	94	27.48
Video	344KB	11289600	7603200	0	600	48	32.65
Barcodes	9.67MB	10137600	8601600	0	14400	5	15.15
Foreman	324KB	22809600	15510528	9	2358	288	32
Vip concentricity	868KB	816642	622536	2	672	4	23.77
Traffic	204KB	6912000	4807680	0	1820	107	30.44

Fig.4 First four frames of video 'traffic' (original frame)

Fig.5 First four reconstructed frames of video 'traffic'

Fig.6 First four frames of video 'Rhinos' (original frame)

Fig.7 First four reconstructed frames of video 'Rhinos'

7. Conclusion

From the observation we can conclude that the clarity of the compressed video is good and the difference between the original and the compressed video is almost zero. To process the RMB, a motion vector is calculated from only the 64 moving pixels of the best-match pattern. To avoid multiple 8×8 blocks of DCT calculations for 64 residual error values per RMB, these 64 values are rearranged into an 8×8 block. A similar inverse rearrangement is performed during the decoding. We proposed an efficient very low bit-rate video coding scheme. Eight pre-defined patterns were chosen by experiments to represent moving regions. Based on the predefined patterns, the computation required for motion estimation can be reduced and better prediction gains can be achieved. In order to reduce the size of a MB to be encoded, we devised a rearrangement method to compact the residual errors of a MB into a block of size 8 × 8. However, the total number of detected RMB will decrease in motion-intensive video sequences, in which its performance will be degraded.

References

[1] Mohammed Ghanbari, *Video Coding and Introduction to Standard Codes.* IEEE Press, 1999.

[2] I. E. G. Richardson, *Video Coding for Next Generation Multimedia.* Hoboken: John Wiley & Sons Ltd.,2003

[3] G. J. Sullivan, P. Topiwala, and A. Luthra, The H.264/AVC Advanced Video Coding Standard: Overview and Introduction to the Fidelity Range Extensions, *Presented at the SPIE Conference on Applications of Digital Image Processing XXVII Special Session on Advances in the New Emerging Standard*: H.264/AVC, August 2004.

[4] Thomas Wiegand, Gary J. Sullivan*,* Overview of the H.264/AVC Video Coding Standard, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol.13, No.7,July 2003, pp. 560-576

[5] Fukuhara, T., K. Asai, and T. Murakami, Very low bit-rate video coding with block partitioning and adaptive selection of two time-differential frame memories, *IEEE Transactions on Circuits and Systems for Video Technology*, 7, pp.212-220, 1997

[6] ISO/IEC N4030, MPEG-4 International Standard, 2001.

[7]. ITU-T Recommendation H.261, .Video codec for audio-visual services at p×64kbits/s,. 1993.

[8] Draft ITU-T Recommendation H.263, .Video coding for low bit-rate communication,. 1996.

[9] Draft ITU-T Recommendation H.263, .Video coding for low bitrate communication. Version 2, 1998.

[10] ITU-T Q6/SG16 VCEG-L45, .H.26L Test Model Long Term Number 6 (TML-6) draft 0. March 2001.

[11] Paul, M., M. Murshed, and L. Dooley, A Variable Pattern Selection Algorithm with Improved Pattern Selection Technique for Low Bit-Rate Video-Coding Focusing on Moving Objects,. Proc. of Int. Workshop on Knowledge Management Technique, Crema, Italy, September 16.18, Vol-2, 1560-1564, 2002.

[12] Wong, K.-W., K.-M. Lam, and W.-C. Siu, An Efficient Low Bit-Rate Video-Coding Algorithm Focusing on Moving regions,. *IEEE Transactions on Circuits and Systems for Video Technology*, Vol 11 No. 10, pp. 1128-1134, 2001.

[13] Paul, M., M. Murshed, and L. Dooley, A Low Bit-Rate Video-Coding Algorithm Based Upon Variable Pattern Selection. *Proc. of 6th Int. Conf. on Signal Processing,* Beijing, China, August 26.30, Vol-2, pp. 933-936, 2002.

[14] Paul, M. and Murshed, M. Advanced Very Low Bit Rate Video Coding Using Preferential Pattern Selection Algorithms, *Journal of Research and Practice in Information Technology*, Vol. 37, No. 3, August 2005.