International Journal of Electronics Communication and Computer Engineering
Volume 4, Issue (6) NCRTCST-2013, ISSN 2249–071X

National Conference on Recent Trends in Computer Science and Technology (NCRTCST)-2013

Video Compression Based on RMB

P. Ushashree
Assistant Professor, Department of Computer Science and Engg.,
Geethanjali College of Engineering & Technology
Cheeryal, Hyderabad
Email: ushashree.sgs@gmail.com

Preeti Prasada
Assistant Professor, Department of Information Technology,
Geethanjali College of Engineering & Technology
Cheeryal, Hyderabad
Email: preeti.prasada@yahoo.com

***Abstract* – In this paper a technique is used to compress the video clips, so that it is a low bit rate video. The selection of an optimal regular-shaped pattern set for very low bit-rate video coding, focusing on moving regions has been the objective of much recent research in order to try and improve bit-rate efficiency. If a particular block does not allow for accurate motion compensation, then it is split into two using the horizontal or vertical line that achieves the maximum reduction in motion compensation error. This method causes partitioning to occur along motion boundaries, thus substantially reducing blocking artifacts. In addition, small blocks are placed in regions of complex motion, while large blocks cover regions of uniform motion. The paper implements Exhaustive Search which is a types of block matching algorithm**.

***Keywords* – Block Matching, Motion Estimation, Video Compression, MPEG, H.261, H.263, H.264**

I. Introduction

Now a days, the demand for applications of the digital video communication, has increased to a large scale. However, the transmission rates over public switched telephone networks (PSTN) are very limited. Therefore, very low bit-rate video coding is an important technology for such applications. ITU-T recommendation H.263 and H.263+ are the successful international standards for video compression using block-based techniques. In these standards, motion estimation and compensation are used to reduce temporal redundancies, and discrete cosine transform (DCT) is then applied to encode the motion-compensated prediction difference.

Reducing the transmission bit-rate while concomitantly retaining image quality is the most daunting challenge to overcome in the area of very low bit-rate video coding, e.g., H.26X standards [3].[6]. The MPEG-4 [2] video standard introduced the concept of content-based coding, by dividing video frames into separate segments comprising a background and one or more moving objects. This idea has been exploited in several low bit-rate macroblock-based video coding algorithms [1][8] using a simplified segmentation process which avoids handling arbitrary shaped objects, and therefore can employ popular macroblock-based motion estimation techniques. Such algorithms focus on moving regions through the use of regular pattern templates, from a pattern codebook (Figure 1), of non-overlapping rectangular blocks of 16×16 pixels, called macroblocks (MB).

The algorithm proposed by Wong et al [8], used eight fixed patterns (Fixed 8), with macroblocks classified according to the following three mutually exclusive classes:

1) Static MB (SMB): Blocks containing little or no motion;

2) Active MB (AMB): Blocks that contain moving object(s) with little static background; and

3) Active-Region MB (RMB): Blocks that contain both static background and some part(s) of moving object(s).

In [1], 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 1). In [9], Paul et al. presented a Variable Pattern Selection (VPS) algorithm to select the λ best-matched patterns from a codebook of patterns P_1.P_{24} in Figure 1 using a .greedy. algorithm, where $\lambda \in \{4,8,16,24\}$. The most computationally expensive and resource hungry operation in the entire compression process is motion estimation. Hence, this field has seen the highest activity and research interest in the past two decades. This paper implements and evaluates the fundamental block matching algorithms from the mid-1980s up to the recent fast block matching algorithms of year 2002.

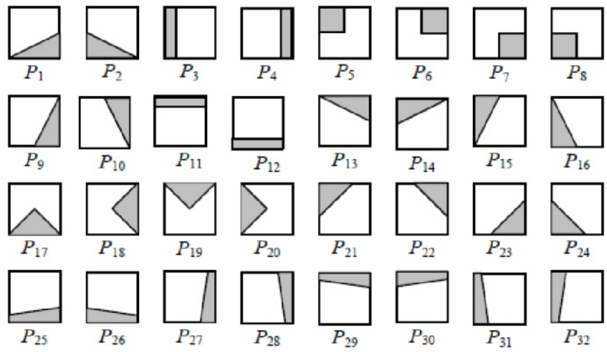

Fig.1. The pattern codebook of 32 regular shaped, 64-pixel patterns, defined in 16×16 blocks, where the shaded region represents 1 (motion) and white region represents 0 (no motion).

The algorithm that has been implemented is Exhaustive Search (ES). Motion compensation techniques are an important part of almost all video codecs since they provide an effective way of exploiting the temporal redundancy between frames in an image sequence. Traditionally, Fixed-Size Block Matching (FSBM) has

All copyrights Reserved by NCRTCST-2013, Departments of CSE
CMR College of Engineering and Technology, Kandlakoya(V), Medchal Road, Hyderabad, India.
Published by IJECCE (www.ijecce.org)

been used to determine the motion of each block in the current frame relative to the reference frame(s).

The paper is organized as follows. The video coding strategy using variable patterns to represent moving regions is described in Section 2. Section 3 presents the video coding strategy using VSBM to represent moving regions. Block matching algorithm used for motion estimation in video compression is described in Section 4 , while simulation results are discussed in Section 5. Section 6 concludes the paper.

Many of the video coding standards tend to employ block-based techniques because of their implementation simplicity and also because they generally provide good results when the bandwidth requirement is relaxed e.g., in MPEG-1/2. This is however, not the case with low bit-rate block-based video coding such as in H.263. The shape of a moving object is generally arbitrary and may not necessarily be aligned with the hypothetical grid structure created by the fixed-sized, non-overlapping rectangular blocks, termed macroblock (MB) in the coding standards. The typical size of a MB being 16×16 pixels, which leads to a large number of blocks, some of which will contain only static background, some will have moving objects and some a combination of the two. In [11], macroblocks were classified according to the following three mutually exclusive classes:

i) *Static MB (SMB)*—Blocks that contain little or no motion;
ii) *Active MB (AMB)*—Blocks that contain moving object(s) with little static background;
iii) *Active-Region MB (RMB)*—Blocks that contain both static background and some part(s) of moving object(s).

By treating AMB and RMB alike, as is done in H.263/H.263+, leads to coding inefficiencies [1]. In order to improve this efficiency, block size may be reduced only to add additional information to be transmitted due to the increase in the number of blocks [10].

Both [1] and [11] successfully addressed the above issue by segmenting each RMB into two regions using a fixed number of predefined regular patterns. They respectively considered four, 128-pixel and eight, 64-pixel predefined RMB patterns. Once the segmentation process was complete, motion estimation/compensation was then only performed on moving regions. Each SMB was skipped for transmission (since they did not contain motion and could be copied from the reference frame) and each AMB was treated exactly as defined in H.263 standard, using motion estimation and compensation techniques. The non- coding of SMB patterns and limiting the number of RMBs to a prescribed set of patterns lead to an improved coding efficiency.

II. Low Bit-Rate Video Coding using Fixed Patterns

2.1 Moving Region Detection:

The basis of this technique is to let the first eight patterns P_1–P_8 in Figure 1 approximate the moving region. Let $C_k(x,y)$ and $R_k(x,y)$, the block of the current and the reference 0≤x,y≤15 denote the k the frames respectively. The moving region in the k block of the current frame is obtained as follows:

$$M_k(x,y) = T(|\,C_k(x,y) \bullet B - R_k(x,y) \bullet B\,|)$$

where B, a square pattern of size 3×3, is the structuring element of morphological closing operations [2][9], |v| returns the absolute value of v, T(v) returns 1 if v > 2 or 0 otherwise, and , 0≤x,y≤15 .

Each block is then classified into SMB, AMB, and RMB according to the following rules. For the k-th block, if 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. 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:

$$D_{k,n} = \frac{1}{256}\sum_{x=0}^{15}\sum_{y=0}^{15}|\,M_k(x,y) - P_n(x,y)\,|$$

Where, 1≤x,y≤32. A candidate RMB is classified as an RMB if min($D_k(x,y)$) <0.25; otherwise it is an AMB.

2.2 Motion Estimation and Compensation

Since both each 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.

III. Motion Compensation Error Surfaces

In traditional block matching, the goal is to minimize the error between a block in the current frame and a displaced block in the reference frame. The distortion is usually measured in terms of either the sum of absolute error (SAE) or the sum of squared error (SSE). In effect, motion estimation amounts to finding the location of the minimum value on the error surface. Because they will prove useful later (in Sections 6.4 and 6.5), the process of generating motion compensation error surfaces is discussed below in more detail. For each block, an error surface Eb,f (u, v) is calculated, i.e. the SSE when block

All copyrights Reserved by NCRTCST-2013, Departments of CSE

International Journal of Electronics Communication and Computer Engineering
Volume 4, Issue (6) NCRTCST-2013, ISSN 2249–071X

number b is motion compensated by translating the corresponding block in reference frame number f a distance (u, v). For all points (x, y) in block b, the SSE between the current frame, I, and a reference frame, If, is calculated according to the equation:

$$E_{b,f}(u,v) = \sum_{(x,y) \in \text{Block } b} [I(x,y) - I_f(x+u, y+v)]^2$$

For a given block, the translational motion vector (measured to pixel accuracy) can be found by determining the vector (u, v) that minimizes Eb,f (u, v). If required, sub-pixel motion can be estimated — either by performing matching (using interpolated pixel values at non-integer positions) or by interpolating the location of the minimum value on the error surface. In general, it is possible to use multiple reference frames for block matching. For each block, it is necessary to calculate one error surface per reference frame.

IV. Block Matching Algorithm

The underlying supposition behind motion estimation is that the patterns corresponding to objects and background in a frame of video sequence move within the frame to form corresponding objects on the subsequent frame.

The idea behind block matching is to divide the current frame into a matrix of ‗macro blocks' that are then compared with corresponding block and its adjacent neighbors in the previous frame to create a vector that stipulates the movement of a macro block from one location to another in the previous frame. This movement calculated for all the macro blocks comprising a frame, constitutes the motion estimated in the current frame. The search area for a good macro block match is constrained up to *p* pixels on all fours sides of the corresponding macro block in previous frame.

This ‗*p*' is called as the search parameter. Larger motions require a larger *p*, and the larger the search parameter the more computationally expensive the process of motion estimation becomes. Usually the macro block is taken as a square of side 16 pixels, and the search parameter *p* is 7 pixels. The matching of one macro block with another is based on the output of a cost function. The macro block that results in the least cost is the one that matches the closest to current block. There are various cost functions, of which the most popular and less computationally expensive is Mean Absolute Difference (MAD) given by equation (i). Another cost function is Mean Squared Error (MSE) given by equation (ii).

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

$$MSE = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \left(C_{ij} - R_{ij} \right)^2$$

where N is the side of the macro bock, Cij and Rij are the pixels being compared in current macro block and reference macro block, respectively. Peak-Signal-to-Noise-Ratio (PSNR) given by equation (iii) characterizes the motion compensated image that is created by using motion vectors and macro clocks from the reference frame.

4.1. Exhaustive Search (ES)

This algorithm, also known as Full Search, is the most computationally expensive block matching algorithm of all. This 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 highest PSNR amongst any block matching algorithm. Fast block matching algorithms try to achieve the same PSNR doing as little computation as possible. The obvious disadvantage to ES is that the larger the search window gets the more computations it requires.

V. Results

Exhaustive Search algorithm has been implemented. ‗Caltrain 'video sequence with a distance of 2 between current frame and reference frame was used to generate the frame-by-frame results of the algorithms. Using eight patterns instead of four patterns not only improves PSNR but also classifies more blocks as RMBs, which contributes towards higher coding compression even after compensating for the larger codebook size (one extra bit per RMB) requirement. As a moving region covers part of an object, the region must start from the edge of the boundary. The moving region must be a convex polygon so that it is simple and regular.

All the RMBs that were initially matched against a pattern, outside the selected patterns, should be considered as candidate RMBs to be matched against the selected patterns. Some of these candidate RMBs may not be classified as RMBs and the frequency of the patterns may also be changed. In some cases, this change may lead to a different ordering in the optimal pattern set.

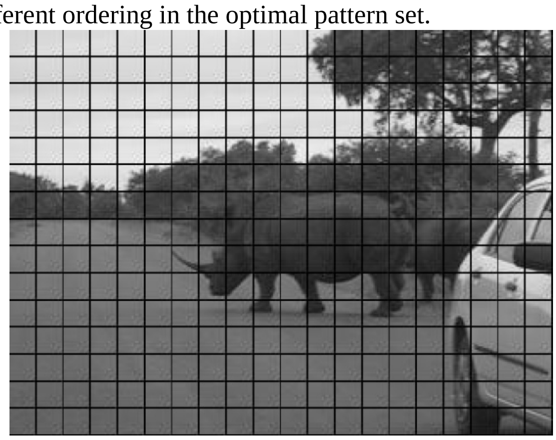

Fig.2. FSBM of size 16×16 with total no.of blocks 3000.

All copyrights Reserved by NCRTCST-2013, Departments of CSE
CMR College of Engineering and Technology, Kandlakoya(V), Medchal Road, Hyderabad, India.
Published by IJECCE (www.ijecce.org)

International Journal of Electronics Communication and Computer Engineering
Volume 4, Issue (6) NCRTCST-2013, ISSN 2249–071X

National Conference on Recent Trends in Computer Science and Technology (NCRTCST)-2013

VI. Conclusion

In this paper a technique of video compression is implemented where the method focuses on moving regions. Each frame is divided in to many macroblocks, using this the moving regions are detected and the compression is done in an efficient manner.

Block matching techniques are the most popular and efficient of the various motion estimation techniques. This paper first describes the motion compensation based video compression in brief. It then illustrates the most popular block matching algorithm.

References

[1] 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 Trans. Circuits Syst. Video Techno., 7, 212.220, 1997

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

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

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

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

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

[7] 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.

[8] Wong, K.-W., K.-M. Lam, and W.-C. Siu, .An Efficient Low Bit-Rate Video-Coding Algorithm Focusing on Moving regions,. IEEE trans. circuits and systems for video technology, 11(10), 1128.1134, 2001.

[9] Paul, M., M. Murshed, and L. Dooley, .A Low Bit-Rate Video-Coding Algorithm Based Upon Variable Pattern Selection,.th Proc. of 6 Int. Conf. on Signal Processing, Beijing, China, August 26.30, Vol-2, 933.936, 2002.

[10] M. P. Servais, T. Vlachos, and T. Davies, Progressive Polygon Encoding of Segmentation Maps, in *Proc. ICIP*, 2004.

[11] Renxiang Li, Bing Zeng, and Ming L. Liou, —A New Three-Step Search Algorithm for Block Motion Estimation, *IEEE Trans. Circuits And Systems For Video Technology*, vol 4., no. 4, pp. 438-442, August 1994.

[12] Jianhua Lu, and Ming L. Liou, A Simple and Efficent Search Algorithm for Block-Matching Motion Estimation, *IEEE Trans. Circuits And Systems For Video Technology*, vol 7, no. 2, pp. 429-433,

All copyrights Reserved by NCRTCST-2013, Departments of CSE
CMR College of Engineering and Technology, Kandlakoya(V), Medchal Road, Hyderabad, India.
Published by IJECCE (www.ijecce.org)