A low bit-rate video-coding algorithm based upon variable pattern selection

Conference or Workshop Item

A LOW BIT-RATE VIDEO-CODING ALGORITHM BASED UPON VARIABLE PATTERN SELECTION

Manoranjan Paul, Manzur Murshed, and Laurence Dooley

Gippsland School of Computing and Info. Tech., Monash University, Churchill Vic 3842, Australia
E-mail: {*Manoranjan.Paul,Manzur.Murshed,Laurence.Dooley*}*@infotech.monash.edu.au*

ABSTRACT

Recent research into pattern representation of moving regions in blocked-based motion estimation and compensation in video sequences, has focused mainly upon using a fixed number of regular shaped patterns. These are used to match the *macroblocks* in a frame that have two distinct regions involving static background and moving objects. In this paper a new Variable Pattern Selection (VPS) algorithm is presented which selects a preset number of best-matched patterns from a pattern codebook of regular shaped patterns. While more patterns are used than in the previous work, the performance of the VPS algorithm in using variable length coding, by exploiting the frequency of the best-matched patterns, leads to a higher compression ratio, without degrading the overall image quality.

Keywords—*Motion estimation and compensation, low bit-rate video coding, moving region detection, pattern matching.*

1 INTRODUCTION

Researchers into very low bit-rate digital video coding are often faced with the daunting challenge of meeting two diametrically conflicting requirements—reducing the transmission bit-rate while concomitantly retaining image quality. Computational complexity and system bandwidth limitations of a communication media mean these two factors tend to be inversely proportional. H.261 [6], H.263 [7], H.263+ [8], MPEG-1/2 [3][4], MPEG-4 [5] are some of the well-known contemporary standards for video compression. H.263 and H.263+ for example, are widely used in video-telephony and video-conferencing applications, where very low bit-rates to accommodate public switched telephone networks (PSTN) is required

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) ***S*tatic *MB* (SMB)**—Blocks that contain little or no motion;
ii) ***A*ctive *MB* (AMB)**—Blocks that contain moving object(s) with little static background;
iii) **Active-*R*egion *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.

Using eight instead of four patterns improved the *peak signal to noise ratio* (PSNR) by up to 0.6 dB [11]. It also meant that better classification of the RMB blocks was achieved, thus contributing to a higher compression ratio, even after compensating for the larger size (3-bit) codebook required for RMB coding. This paper presents an extension of this concept by proposing a new Variable Pattern Selection (VPS) algorithm, using 24 regular shaped patterns. The VPS algorithm selects a preset number of the best-matched patterns from the 24 available patterns. The best-match pattern selection process is based upon the matching frequency of each pattern. These frequency values are used to code pattern identifier numbers using variable length coding techniques including Huffman and arithmetic coding. This leads to an improvement in coding efficiency. The VPS algorithm exhibits both better compression and PSNR performance compared with the algorithm using a fixed number of patterns [11] (see Section 2)

This paper is organized as follows. In Section 2, the low bit-rate video-coding algorithm focusing on moving region using fixed patterns [11] is explained, while the new VPS algorithm that uses best-matched patterns from a relatively large pattern codebook is developed in Section 3. Section 4 provides a number of performance evaluation results based on simulation and Section 5 concludes the paper.

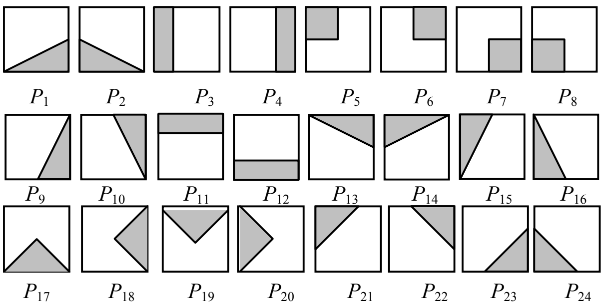

Figure 1: 24 regular shaped 64-pixel patterns, defined in 16×16 blocks, where the shaded region represents 1's and the white region represents 0's.

2 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 $R_k(x,y)$, $0 \le x,y \le 15$, denote the k^{th} block of the current and the reference frames respectively. The moving region $M_k(x,y)$ in the k^{th} block of the current frame is obtained as follows:

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

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 $0 \le x,y \le 15$.

Each block is then classified into SMB, AMB, and RMS according to the following rules. For the k-th block, if $M_k(x,y)$ 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 eight prescribed patterns and the best-match pattern is obtained by minimizing the following expression: -

$$D_{k,n} = \frac{1}{256}\sum_{x=0}^{15}\sum_{y=0}^{15}|M_k(x,y) - P_n(x,y)| \quad (2)$$

where $1 \le n \le 8$. A candidate RMB is classified as an RMB if $\min(D_{k,n}) < 0.25$; 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.

2.3 Encoding and Decoding

Having dealt with the coding/decoding of SMB and AMB, the issue of how to process the RMB blocks is now discussed. 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 only 64 residual error values per RMB, these 64 values are *rearranged* into an 8×8 block. Similarly, an *inverse rearrangement* is performed decoding.

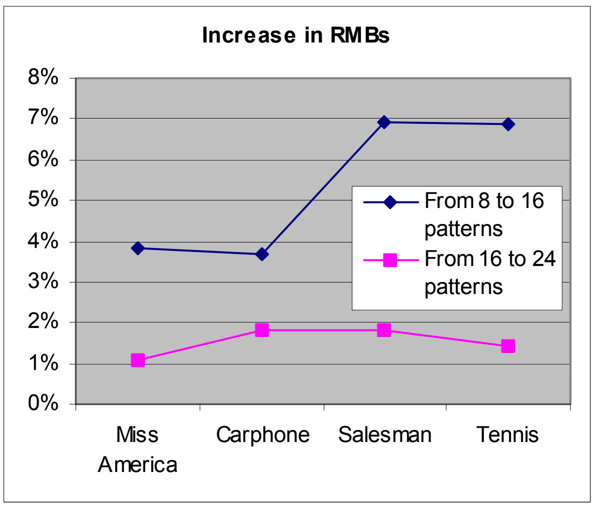

Figure 2: Percentage increase in RMBs when the number of patterns is increased from 8 to 16 and from 16 to 24.

3 LOW BIT-RATE VIDEO CODING USING VARIABLE PATTERNS

As stated in Section 1, the attraction of using more than eight patterns comes from the observation in [11] that 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. However, as Figure 2 clearly shows for a selection of popular video sequences, the increase in RMB becomes insignificant once the number of patterns is greater than 16.

Another interesting observation is made in Figure 3, which shows that the maximal-frequency pattern sequence is not fixed for all types of video data. For example, the most frequent eight patterns for "Miss America" video sequence is 6, 5, 7, 4, 8, 3, 1, and 16 (in order); whereas the same for "Tennis" video sequence is 5, 6, 8, 7, 11, 1, 3, and 2 (in order). It confirms the general judgment, that the first eight patterns in Figure 1, which were used in [11], are not the optimal set.

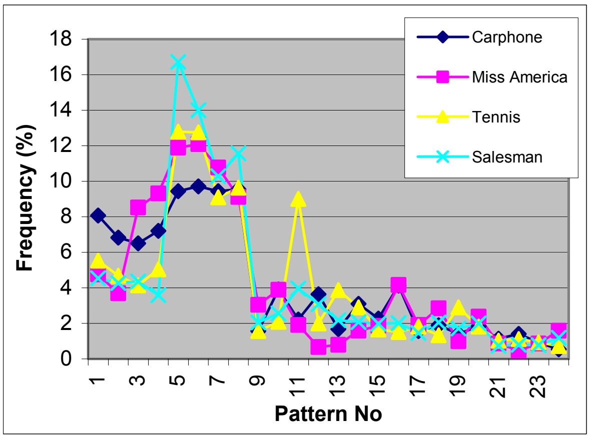

Figure 3: Frequency of patterns for selective video sequences.

It is therefore, a reasonable objective to use the maximal-frequency pattern set from a larger pattern bank.

3.1 Proposed Patterns

In our new approach, 24 regular shaped 64-pixel patterns in Figure 1, defined in 16×16 blocks, are used where the shaded region represents 1 and the white region represents 0. The patterns are selected intuitively based on the following two features:

- 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.

3.2 Variable Pattern Selection (VPS) Algorithm

The principles of this algorithm are to select the optimal pattern set of a given size from a pattern codebook (the 24 patterns in Figure 1). However, selecting such an optimal set is not straightforward, as for example, in finding the optimal set of eight patterns, it is not sufficient to simply select the patterns with the highest frequencies as given in Figure 3. 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. For example, the order of the most frequent sixteen patterns for "Miss America" video sequence in Figure 3 is 6, 5, 7, 4, 8, 3, 1, 16, 10, 2, 9, 18, 20, 11, 17, and 15; whereas the same in Figure 4 (the optimal one) is 5, 7, 6, 8, 4, 3, 10, 16, 9, 11, 18, 20, 1, 2, 15, and 17. The VPS algorithm, therefore, must eliminate the least frequent pattern in each iteration in order to obtain the optimal pattern set for a given size. The full VPS algorithm is formalised in Figure 5.

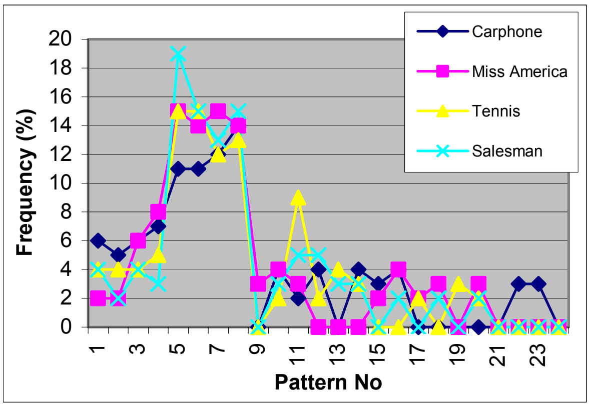

Figure 4: Frequency of optimal sixteen patterns for selective video sequences.

Algorithm VPS(S)

Parameter: S = Size of the optimal pattern set.

Return: O = The optimal pattern set.

Step 1: $O = \{P_1, P_2, \ldots, P_{24}\}$;

Step 2: if $|O| = S$ then exit;

Step 3: Calculate the frequency of each pattern in O;

Step 4: Find the pattern $P_i \in O$ such that its frequency is the minimum;

Step 5: $O = O - \{P_i\}$;

Step 6: Go back to Step 2;

Figure 5: The VPS algorithm.

Discarding only a single pattern during each iteration is expensive, especially when the optimal pattern set size is low. However, this process can be improved by discarding more than one pattern, depending on the relative frequency.

An additional advantage to this approach is that the frequency information can now be used to code pattern identifier numbers using a variable length coding, such as Huffman coding. For example, instead of using fixed codebook lengths of 4 bits, the optimal sixteen pattern ifor "Miss America" video sequence can be Huffman coded using on average only 3.62 bits.

4 EXPERIMENTAL RESULTS

Both the new VPS algorithm and algorithm presented in [11] have been applied to a number of standard and non-standard video sequences to generate coded files. VPS algorithm produced an improvement of 4 between 7% more RMBs (see Figure 2) with 3 to 8% less pattern matching error ($D_{k,n}$) compared with the performance of the algorithm in [11]. The increase in RMBs and use of Huffman coding for identifying patterns are translated directly in achieving increased compression as high as 2.84% (see Table I). VPS algorithm, on average, saved over 300 bytes per QCIF type frame in raw data, while exhibiting no degradation in PSNR as evidenced in Table II.

Table I: Relative compression by VPS(16) algorithm compared with the algorithm in [11].

	Picture Format	Reduction in coded file size
Miss America	QCIF	2.84%
Carphone	QCIF	0.76%
Salesman	CIF	1.41%
Tennis	CIF	0.69%

Table II: PSNR comparison.

	Picture Format	Method in [11]	VPS(16)
Miss America	QCIF	42.80	42.85
Carphone	QCIF	38.83	38.87
Tennis	CIF	36.60	36.61
Salesman	CIF	38.79	38.74

5 CONCLUSION

Recently several studies on pattern representation of moving regions in blocked-based video motion estimation and compensation have been reported. All the studies however have used only a fixed number of regular shaped patterns for all video sequences, to match those macro blocks that have two distinct regions, namely those involving static background and moving objects. In this paper a new *Variable Pattern Selection* (VPS) algorithm is presented which selects a preset number of best-matched patterns from a pattern codebook of 24 regular shaped patterns. Experimental results have shown the importance of using an optimal set of patterns, that varies with video sequences, rather a fixed pattern set. The VPS algorithm captures more RMBs due its use of an optimal pattern set. An added benefit lies in its ability to use variable length coding, by exploiting the frequency of the best-matched patterns. These benefits are successfully translated into achieving better compression ratio without degrading the overall *peak signal to noise ratio* in experiments involving a number of standard and non-standard video sequences.

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 Technol.*, 7, 212–220, 1997.

[2] Gonzalez, R.C. and R. E. Woods, *Digital Image Processing*, Addition-Wesley, 1992.

[3] ISO/IEC 11172, International Standard, 1992.

[4] ISO/IEC 13818, MPEG-2 International Standard, Video Recommendation ITU-T H.262, 1995.

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

[6] ITU Telecom. Standardization Sector of ITU, "Video codec for audio-visual services at *p*× 64kbits/s," ITU-T Recommendation H.261,1993.

[7] ITU Telecom. Standardization Sector of ITU, "Video coding for low bitrate communication," Draft ITU-T Recommendation H.263, 1996.

[8] ITU Telecom. Standardization Sector of ITU, "Video coding for low bitrate communication," Draft ITU-T Recommendation H.263, Version 2, 1998.

[9] Maragos, P., "Tutorial on advances in morphological image processing and analysis," *Opt. Eng.*, 26(7), 623–632, 1987.

[10] Shi, Y.Q. and H. Sun, *Image and Video Compression for Multimedia Engineering Fundamentals, Algorithms, and Standards*, CRC Press, 1999.

[11] 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.