Advanced Very Low Bit Rate Video Coding Using Preferential Pattern Selection Algorithms

Manoranjan Paul and Manzur Murshed

Gippsland School of Computing and IT,
Monash University, Churchill, Vic 3842, Australia
Email: Manoranjan@ieee.org, Manzur.Murshed@infotech.monash.edu.au

In the context of very low bit-rate video coding, pre-defined fixed pattern representations of moving regions in blocked-based motion estimation and compensation has become increasingly attractive over H.264 as the former represents an MB by a smaller size moving region covered by the best available pattern that approximates the shape of the region more closely and hence, requiring no extra motion vector, which is not the case with the latter. But fixed set patterns sometimes fail to code efficiently for all video sequences. In this paper a novel idea of selecting a subset of best-matched patterns through preferential selection technique is developed by presenting two algorithms, Variable Pattern Selection *(VPS) and* Extended VPS *(EVPS) for an initial pattern codebook size of 32 using a new parametric macroblock classification definition and a new similarity metric. The complexity analysis confirmed that EVPS guaranteed to be nearly six times faster than VPS, with the peak performance providing an improvement factor of nine times. The overall performance of EVPS is identical to VPS for certain parameters but on average, 0.2dB and 0.8dB better than the contemporary algorithm using fixed set patterns and Advanced Video coding standard (H.264) respectively, for the same number of bits per frame.*

ACM Classification: I.4.2 (Computing methodologies – image processing and computer vision – compression (coding))

1. INTRODUCTION

Reducing the transmission bit-rate while concomitantly retaining image quality continues to be a major challenge for efficient very low bit-rate video compression standards, such as H.26X (ITU-T Rec. H.264, 2003; ITU-T Rec. H.261, 1993; ITU-T Rec. H.263, 1996; 1998; 2000). These standards however, are unable to encode moving objects within a 16×16 pixel *macroblock* (MB) during *motion estimation* (ME), resulting in all 256 residual error values being transmitted for *motion compensation* (MC), regardless of whether there are moving objects or not. This issue was first addressed by Fukuhara *et al* (1997) who used four MB-partitioning patterns each comprising 128-pixels. ME and MC were applied to all eight possible 128-pixel partitions of an MB and the pattern with the smallest prediction error selected. Having only four patterns however, meant it was insufficient to represent moving objects (Wong *et al*, 2001).

The MPEG-4 (ISO/IEC 14496, 1998) video standard first introduced the concept of content-based coding, by dividing video frames into separate segments comprising a background and one or more moving objects. Wong *et al* (2001) further exploited the idea of partitioning the MBs, via

Manuscript received: 17 May 2004
Communicating Editor: Robyn Owens

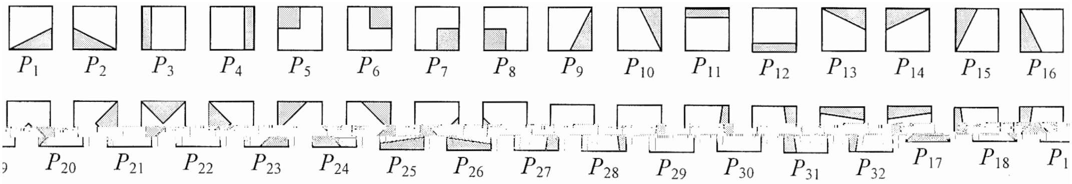


Figure 1: The codebook of 32 regular shaped 64-pixel patterns, defined in 16×16 blocks, where the shaded region represents 1 (motion) and white region represents 0 (no motion).

a simplified segmentation process that avoided handling the exact shape of the moving objects, so popular MB-based ME techniques could be applied. This technique focused on the moving regions of the MBs by using a *pattern codebook* (PC) comprising eight, regular 64-pixel patterns (P_1–P_8 in Figure 1). If the moving region of a MB is well covered by a particular pattern, then the MB can be coded by considering only the 64 pixels of that pattern, with the remaining 192 pixels being skipped as static background. Wong *et al* (2001) classified each MB into three mutually exclusive categories: 1) *Static MB* (SMB): MBs containing little or no motion; 2) *Active MB* (AMB): Blocks containing moving object(s) with little static background and 3) *Active-Region MB* (RMB): Blocks containing both static background and part(s) of moving object(s) represented by one pattern from the PC. The first two MB types are defined in the H.26X standard (ITU-T Rec. H.264, 2003; ITU-T Rec. H.261, 1993; ITU-T Rec. H.263, 1996; 1998; 2000) and treated in exactly the same way, while for the RMB type, ME and MC is performed only for those moving regions covered by a selected pattern from the codebook. Overall, this affords superior prediction and compression efficiency as well as reducing the encoding time compared to H.263 by between 8% and 53% for smooth motion sequences (Wong *et al*, 2001).

In implementing the above categories, a MB is considered as a *candidate RMB* (CRMB) if at least one of the four 8×8 quadrants does not possess moving pixels (Wong *et al*, 2001). This classification may, in certain instances reduce the number of RMBs by misclassifying a possible CRMB as an AMB, where only a few moving pixels exist in another quadrant. Conversely, it may also increase the computational complexity by misclassifying an AMB as a CRMB where all but one quadrant has many moving pixels. Ultimately a CRMB is classified as an RMB depending on a similarity measure with the patterns in the codebook. This paper introduces a new efficient parametric ($\delta \in \{64,96,128\}$) MB classification definition, where δ is the total number of moving pixels in a MB, without regard to any 8×8 quadrant.

Using only eight patterns for all video types, results in potentially many RMBs being neglected as moving regions vary widely between sequences. If the codebook size is extended, the number of RMBs will increase so that for a *Fixed-λ algorithm*†, the image quality improves as the residual error is reduced; though there will be a corresponding increase in the number of pattern identification bits for each RMB. The rationale for extending the PC size and then selecting a subset of patterns based on video content, is that in Wong *et al* (2001) it was observed that the coding efficiency with eight patterns (*Fixed*-8 algorithm) was superior to using only the first four patterns. In this paper, a similar trend but with diminishing returns is observed when the PC is extended to 32 patterns (see Figure 1).

To counter the diminishing improvement in coding efficiency caused by the increased number of bits to identify each of the 32-patterns, only the λ (< 32) best-matched patterns are used. The

† Throughout this paper, any pattern selection algorithm using the same set of λ patterns for a video sequence is termed as *Fixed*-λ algorithm. The eight-pattern algorithm (Wong *et al*, 2001) is, therefore, referred to as the *Fixed*-8 algorithm.

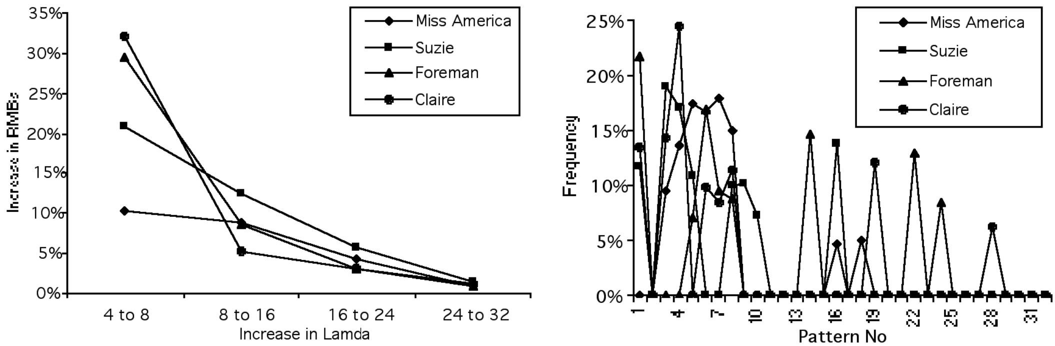


Figure 2: (a) Plot of the number of RMBs vs. pattern number λ; (b) Frequences of the best eight patterns

pattern set selection process is however not straightforward, since in locating the best λ pattern set, it is not sufficient to simply select the patterns with the highest frequencies. All the RMBs that were initially matched against a pattern and subsequently discarded need to be considered again for matching. Some of these patterns may no longer be classified as RMBs and the frequencies of the patterns may also change. In certain cases, this change will lead to a different ranking in the best pattern set. In this paper, a new *variable pattern selection* (VPS) algorithm is developed to select the λ best-matched pattern set using a *preferential selection* approach analogous to the Australian Preferential Voting System (Preferential Voting in Australia), where the pattern with the lowest frequency, i.e., the least preferred pattern by all the RMBs, is eliminated iteratively. As a consequence, all the RMBs originally preferring the eliminated pattern are forced to opt for their next preferred patterns in the following iteration. A second preferential selection algorithm called *extended VPS* (EVPS) is also presented which offers significant processing speed improvements. It will be shown that EVPS reduces the computational complexity by a factor as great as nine while maintaining similar quality and compression efficiency to VPS.

The rest of the paper is organized as follows. Section 2 discusses three key pattern selection issues, which are integral to the new approach, while Section 3 details fully both the VPS and EVPS preferential selection algorithms. The computational complexity of VPS and EVPS is comprehensively analysed in Section 4, while Section 5 provides a performance comparison. Section 6 concludes the paper.

2. THREE PATTERN SELECTION ISSUES

Any absolute performance evaluation based on the video data is impossible because of the diversity of the content and motion patterns that cannot be represented with any predefined correlation function. As a result, coding efficiency of video coding algorithms is compared on the basis of their relative empirical performances on standard video sequences covering different degrees of object and camera motions. In this paper, seven standard video sequences, such as *Miss America* ("talking heads" motion), *Suzie* (hand and head movements), *Mother&Daughter* (two heads movements), *Carphone* (fast object motion with part of back ground), *Foreman* (object translation and panning), *Salesman* (head-shoulder-hand movements), and *Claire* ("talking head" motion) are used.

2.1 Pattern Codebook (PC) Size

From the empirical results in Figure 2(a), it can be observed that for each video sequence, while the number of RMBs monotonically increases with λ, the rate of increase however decreases, which is typically only 1% when the codebook size is increased from 24 to 32 patterns. In this paper, an initial codebook size of 32 patterns (Figure 1) is used, where each 64-pixel pattern is regular – bounded by straight lines, clustered – the pixels are connected, and boundary-adjoined. Another interesting observation may be drawn from Figure 2(b), which shows that the maximal-frequency pattern sequence is not the same for each video sequence. For example, the eight most frequent patterns (ranked) for the "*Miss America*" sequence are 7, 5, 6, 8, 4, 3, 18, and 16; whereas for the "*Foreman*" sequence it is 1, 6, 14, 22, 7, 8, 24, and 5. This proves that using only eight patterns as in the Fixed-8 algorithm, by no means represents the best set for all video sequences.

2.2 Pattern Similarity Metric

Let $C_k(x,y)$ and $R_k(_{x,y})$ denote the k^{th} MB of the current and reference frames, each of size W pixels × H lines, respectively of a video sequence, where $0 \leq x,y \leq 15$ and $0 \leq \text{k} < W/16 \times H/16$. The moving region $M_k(x,y)$ of the k^{th} MB in the current frame is obtained as follows:

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

where B is a 3×3 unit matrix for the morphological closing operation • (Gonzalez and Woods, 1992; Maragos, 1987), which is applied to reduce noise, and the thresholding function $T(v) = 1$ if $v > 2$ and 0 otherwise. In the definition of $T(v)$, v is compared against a small constant 2 in order to detect major changes in intensity while ignoring any subtle change thus minimising the possibility of macroblock misclassification.

The *similarity* of the k^{th} MB to a pattern P_n can be measured using the following distance: .

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

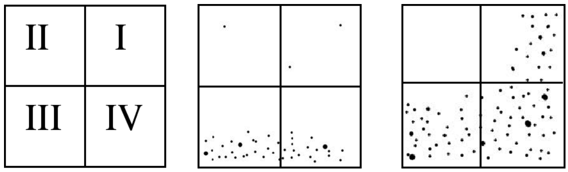
The moving region of the k^{th} MB is best represented by pattern P_n such that

$$D_{k,n} = \min_{\forall P_i \in \text{CPC}}(D_{k,i} \,|\, D_{k,i} < T_S) \tag{3}$$

where T_S is the *similarity threshold*. It is assumed throughout the paper that $T_S = 0.25$, since if none of the 64-pixels of a particular pattern cover any part of a moving region, then the pattern similarity metric will be ≥ 64/256 = 0.25. Note that the patterns and M_k are binary matrix containing only '0' and '1' where '0' represents background and '1' represents foreground in a frame of a video sequence.

2.3 New Macroblock Classification

The MB classification used in Wong's (Wong *et al*, 2001) proposed algorithm has some limitations. In implementing the RMB category, a MB is considered as a CRMB if at least one of the four 8×8 quadrants does not possess moving pixels. This classification may, in certain instances reduce the number of RMBs by misclassifying a possible CRMB as an AMB, where only a few moving pixels exist in another quadrant. Conversely, it may also increase the computational complexity by misclassifying an AMB as a CRMB where all but one quadrant has many moving pixels. Ultimately a CRMB is classified as an RMB depending on a similarity measure with the patterns in the codebook. For example, according to Wong's (Wong *et al*, 2001) definition, the macroblock in Figure 3(b) is not considered as an RMB because of the presence of two '1's and one '1' in quadrants I and II respectively. However, the moving region of this macroblock matches very closely to the pattern P_1 or P_2. On the other hand, the AMB macroblock in Figure 3(c) is matched unnecessarily with the pattern codebook as there is no '1' in quadrant II.



(a) (b) (c)

Figure 3: Example Macroblocks with Quadrants

We introduce a new efficient parametric ($256 > \delta > 64$) MB classification, where δ is the total number of moving pixels in a MB, without regard to any 8x8 quadrant. The justification of the lower and upper limit is that the pattern size is 64 and total maximum moving pixels are 256 respectively. The experimental results show the actual value of δ within $256T_S + 64 > \delta > 64$ under a similarity threshold for VLBR. If δ is more than $256T_S + 64$, then no CRMBs will be classified as RMBs, on the other hand, if δ is less than 64, then some CRMBs will not be classified as RMB but are good candidates to be. Unlike the previous definition, this definition considers the number of '1's in a macroblock irrespective of their position in any specific quadrants. Moreover, this definition introduces a parameter δ, which controls the number of RMBs generated by this definition. In the corollary, parameter δ directly contributes in both quality and compression ratio. The k^{th} MB is coded as follows: IF $\Sigma M_k < 8$ THEN block is an SMB, ELSE IF $\Sigma M_k < \delta$ THEN block is a CRMB, ELSE block is an AMB. IF a CRMB can be represented by a pattern in the λ best-matched pattern set using metric (3) THEN the block is an RMB, ELSE it is an AMB.

3. PREFERENTIAL PATTERN SELECTION ALGORITHMS

3.1 Variable Pattern Selection (VPS) Algorithm

The VPS algorithm, which is detailed in Figure 4(a), selects the λ best-matched pattern set, where $\lambda \in \{4,8,16,24\}$ from the PC using a *preferential selection* approach as follows. Starting with a complete PC of 32 patterns, VPS obtains the λ best-matched pattern set by eliminating the least matched pattern per iteration. The coding is then performed using the λ best-matched pattern set via the Fixed-λ algorithm.

VPS is computationally expensive since for a codebook size of α patterns, $\alpha - \lambda$ iterations are required. Unlike the Fixed-λ algorithm, the VPS algorithm includes in its header the indices of the λ best-matched patterns in the PC, which increases the length of the coded stream by $\lambda\ lg|PC|$ bits. However, this loss in coding efficiency is more than compensated for by using variable length coding (VLC), which exploits the inherent pattern frequencies in VPS, for pattern identification numbering. Empirical results prove that on average, the best eight (sixteen) patterns for seven standard test sequences require only 2.95 (3.82) bits, instead of 3 (4) bits per pattern identification number, when VLC is used. Thus for example, if a sequence generates β RMBs, the coding gain for VLC will be 0.05β bits and 0.18β bits for $\lambda = 8$ and 16 respectively. Normally 20~30% MBs are classified as RMBs in the standard video sequence (Wong *et al*, 2001), which is a significant ratio to improve pattern based coding efficiency.

3.2 Extended VPS (EVPS) Algorithm

To improve the VPS computational efficiency, the *extended VPS* (EVPS) algorithm detailed in Figure 4(b), was proposed, which enables more than one of the least frequent patterns to be eliminated in each iteration. The key premise is that the λ best-matched patterns lie in the first κ ($\geq \lambda$)

Algorithm Ω = VPS(λ)

1: $\Omega = PC$;

2: FOR each RMB, determine the best matched pattern in Ω using (3);

3: FOR each pattern $P_i \in \Omega$, calculate frequency (%), $\text{Freq}(P_i)$, by counting the number of matching RMBs;

4: IF $|\Omega| = \lambda$ THEN exit;

5: Find the pattern $P_i \in \Omega$ such that $\text{Freq}(P_i)$ is the MINIMUM;

6: $\Omega = \Omega - \{P_i\}$;

7: GOTO 2;

(a)

Algorithm Ω = EVPS(λ, γ)

1: $\Omega = PC$;

2: FOR each RMB, determine the best-matched pattern in Ω using (3);

3: FOR each pattern $P_i \in \Omega$, calculate the frequency (%), $\text{Freq}(P_i)$, by counting the number of matching RMBs;

4: IF $|\Omega| = \lambda$ THEN exit;

5: RANK Ω to $(P_{i_1}, P_{i_2}, \ldots, P_{i_{|\Omega|}})$ such that $\text{Freq}(P_{i_j}) \geq \text{Freq}(P_{i_{j+1}})$, for all $j < |\Omega|$;

6: Find the MAXIMUM value of κ such that $\sum_{j=1}^{\kappa} \text{Freq}(P_{i_j}) < \gamma$;

7: $\Omega = \Omega - \{\forall j > \max(\kappa, \lambda) : P_{i_j}\}$;

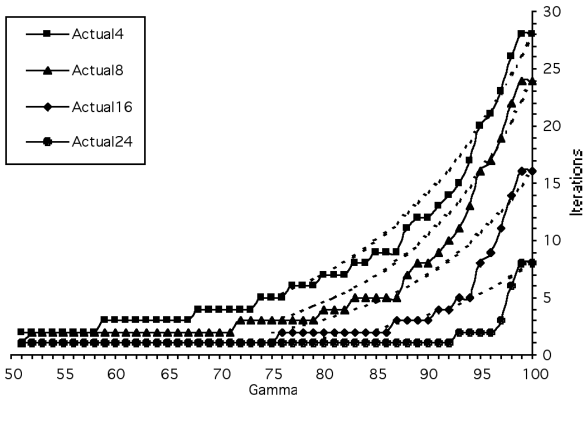
8: GOTO 2;

(b)

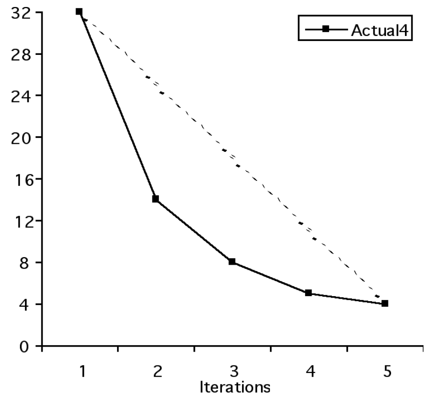
Figure 4: (a) The VPS algorithm; (b) The EVPS algorithm

most frequent patterns, such that their cumulative frequency $\geq \gamma$, where γ is the *pattern elimination confidence factor*. Since in each iteration, the pattern elimination process discards only the least best-matched pattern(s), the confidence factor γ increases with κ. This fast method however does not guarantee that certain best-matched patterns may be amongst the eliminated $\alpha - \kappa$ patterns. It is also interesting to note that for $\gamma = 100\%$, only one pattern per iteration is eliminated, i.e. EVPS(λ, 100%) is equivalent to the VPS(λ) algorithm.

EVPS was applied to seven smooth-motion standard sequences, with *Suzie*, *Mother&Daughter*, *Car Phone* and the *Foreman* sequences generating a higher number of iterations in comparison to



(a)



(b)

Figure 5: For *Suzie* sequence, actual and approximated (dotted lines) values of (a) iterations of EVPS($\lambda \in \{4,8,16,24\}$, $\gamma \in \{51\%,\ldots,100\%\}$) algorithms; (b) intermediate pattern set size of EVPS(4, 75%) algorithm

the *Miss America*, *Salesman*, and *Claire* sequences, since they possess greater motion, especially for low values of γ. The actual and approximated number of iterations of EVPS($\lambda \in \{4,8,16,24\}$, $\gamma \in \{51\%,\ldots,100\%\}$) algorithms for *Suzie* sequence are plotted in Figure 5(a), which reveal the important role γ plays in controlling the number of iterations, and thereby processing speed, especially for small λ. For $\gamma < 75\%$, the benefit in reducing the number of iterations is insignificant, so in order to maintain the pattern elimination confidence level as high as possible, γ will only be considered for the range 75–100% in this paper.

4. COMPUTATIONAL COMPLEXITY ANALYSIS

In this section, the speed up factor for the EVPS algorithm compared with VPS is calculated using both numerical and empirical complexity analysis and by assuming that both algorithms are coding a video sequence with β RMBs. Let the intermediate pattern set at iteration i be denoted as Ω_i for both algorithms. During the i^{th} iteration, Lines 2 and 3 of both algorithms require $\beta|\Omega_i|$ evaluations of $D_{k,n}$ in (2), while keeping track of the minimum distance metric in (3). This operation can be performed in $O(\beta|\Omega_i|)$ time. Line 5 of the VPS algorithm has to identify the least frequent pattern from the current pattern set Ω_i and this is completed in $O(|\Omega_i|)$ time. Line 5 of the EVPS algorithm, however, requires pattern ranking and this takes $O(|\Omega_i|\log|\Omega_i|)$ time, while Line 6 scans the ranked pattern set, which takes $O(|\Omega_i|)$ time.

In both algorithms, the exit condition is at Line 4 in order to exploit information about the frequencies of the final patterns (selected during the previous iteration) that will be used for VLC of pattern numbers. Considering $\alpha - \lambda$ selection iterations and $|\Omega_i| = \alpha - i + 1$, the VPS algorithm can therefore be completed in the following time:

$$\sum_{i=1}^{\alpha-\lambda+1} O(\beta|\Omega_i|) + \sum_{i=1}^{\alpha-\lambda} O(|\Omega_i|) = O\left(\beta \sum_{i=1}^{\alpha-\lambda+1} |\Omega_i| + \sum_{i=1}^{\alpha-\lambda} |\Omega_i|\right) = O\left(\beta \sum_{j=\lambda}^{\alpha} j + \sum_{j=\lambda+1}^{\alpha} j\right) = O\left(\frac{\beta(\alpha+\lambda)(\alpha-\lambda+1)}{2}\right). \quad (4)$$

For the EVPS algorithm, both the number of iterations and size of the intermediate pattern set are functions of γ. From Figure 5(a), the number of iterations at $\gamma = 75\%$ is 5, 3, 1, and 1 for $\lambda = 4$, 8, 16, and 24 respectively. As alluded above, the *Suzie* sequence generates a relatively high number of iterations, so without losing generality, the number of iterations at $\gamma = 75\%$ can be overestimated to be $\lfloor \alpha/2\lambda \rfloor + 1$ for all λ in all sequences. Clearly, the number of iterations at $\gamma = 100\%$ will be the same as VPS i.e. $\alpha - \lambda$, so by applying an exponential curve fit between these two end-points, the number of iterations for any γ is approximated as:

$$\psi = \left(\lfloor \alpha/2\lambda \rfloor + 1\right)\exp\left(4(\gamma - 75\%)\ln\left((\alpha-\lambda)/\left(\lfloor \alpha/2\lambda \rfloor + 1\right)\right)\right). \quad (5)$$

Figure 5(a) shows that the above approximation overestimates the number, in all but the limiting case when $\gamma > 97\%$.

Now consider the intermediate pattern set size. Obviously the initial and final sizes are $|\Omega_1| = \alpha$ and $|\Omega\psi| = \lambda$ respectively. While it has been empirically shown in Figure 5(b) that $|\Omega_i|$ decreases logarithmically with the number of iterations, for the sake of simplicity, the decrease is assumed to be linear:

$$|\Omega_i| = \lambda + (\psi - i)\frac{\alpha - \lambda}{\psi - 1}, \text{ for all } i = 1, 2, \ldots, \psi. \quad (6)$$

Figure 5(b) shows the example of $\lambda = 4$ as it represents the worst case of pattern elimination. All the assumptions made in this analysis overestimate the values concerned to ensure the guaranteed improvement in time complexity achieved by the EVPS algorithm is underestimated. EVPS can therefore be completed in the following time:

$$\sum_{i=1}^{\psi} O(\beta|\Omega_i|) + \sum_{i=1}^{\psi-1} O(|\Omega_i|\log|\Omega_i|) = O\left(\beta\sum_{i=1}^{\psi}|\Omega_i| + \sum_{i=1}^{\psi-1}|\Omega_i|\log|\Omega_i|\right) = O\left(\frac{\psi\beta(\alpha+\lambda)}{2}\right) \quad . \tag{7}$$

It can be concluded that EVPS guarantees a speedup factor of at least $(\alpha - \lambda + 1)/\psi$ computational improvement over VPS. Figure 6 plots the speedup factor for various λ over the entire range of γ. Clearly, the speedup factor is a maximum at $\gamma = 75\%$ and then decreases logarithmically with γ. The analysis also confirms that EVPS guarantees to be at least 5.8 times faster than VPS, with the peak performance achieved for a codebook size of 24 and a *pattern elimination confidence factor* of 75%, when it is a factor of nine times faster in coding all video test sequences.

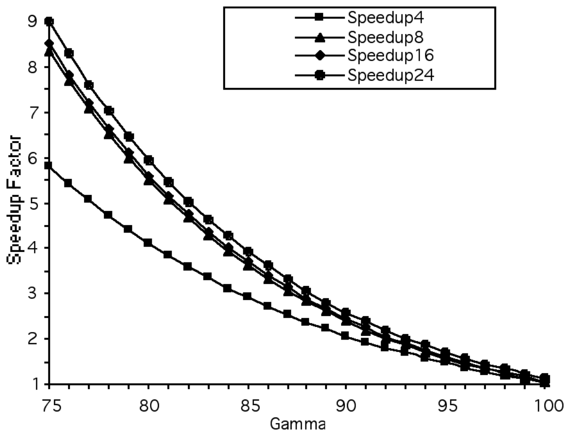


Figure 6: Underestimated speedup factor of the EVPS algorithm with respect to the VPS algorithm

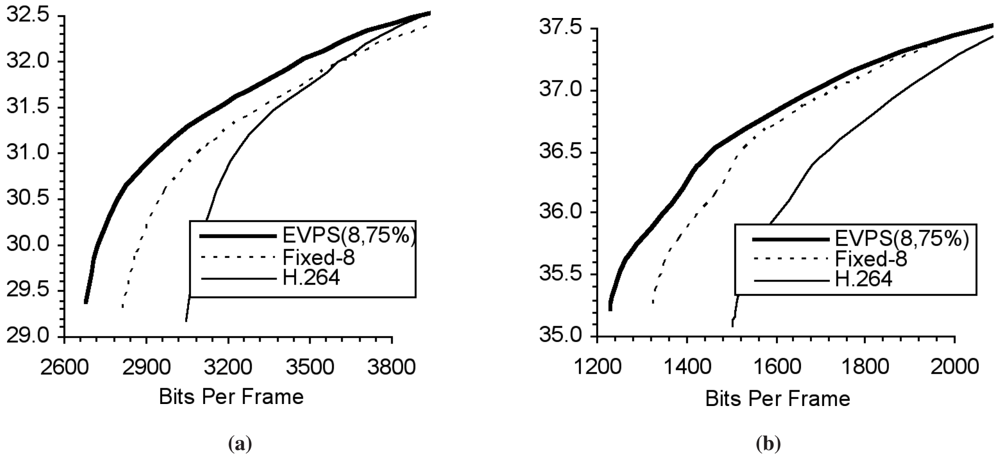


Figure 7: Coding efficiency curves for the *Suzie* (a) and *Miss America* (b) sequences.

Video sequences	Bits per frame	PSNR (dB)			
		H.264	Fixed-8	VPS (8)	EVPS(8, 75%)
Miss America	1500	35.31	36.30	36.70	36.69
Suzie	3100	30.00	30.98	31.17	31.16
Mother& Daughter	2395	28.55	29.96	30.14	30.12
Car phone	3550	29.78	30.11	30.23	30.23
Foreman	4267	28.03	28.32	28.44	28.44
Salesman	2048	30.53	30.96	31.07	31.06
Claire	1024	34.10	34.64	34.84	34.82

Table I: PSNR values for standard sequences using the H.264, Fixed-8, VPS(8), and EVPS(8, 75%) algorithms

5. EXPERIMENTAL RESULTS

The VPS, EVPS, Fixed-8 and H.264 algorithms were implemented in MATLAB 6.1 and their respective coding performance tested on a large number of standard and non-standard video sequences of QCIF digital video format (Shi and Sun, 1999). For brevity all the experimental results are presented using the first 100 frames of seven popular test sequences. The motion estimation involved a full-search, block-matching algorithm with half-pel (Shi and Sun, 1999) accuracy. The variable block-size motion estimation used in H.264 standard is also applied only for AMBs in Fixed-8, VPS and EVPS algorithms. All experiments proved that the actual speedup for the EVPS algorithm, for various λ and γ values, was always much higher than the estimated speedup shown in Figure 6.

To ensure a fair comparison on the quality and compression, only the VPS(8) and EVPS(8, 75%) algorithms are compared with the H.264 standard and Fixed-8 algorithms in Table I. No degradation in quality or compression was found for the EVPS(8, 75%) algorithm compared with the VPS(8) algorithm, with the overall performance on average, being 0.8dB and 0.2dB better than the H.264 standard and Fixed-8 algorithm respectively, for the same bit-rate. Figure 7 (a) and (b) also confirm the superiority of the EVPS algorithm across the entire low bit-rate range for both *Suzie* and *Miss America* sequences. Similar results were obtained for all other standard test sequences analysed. Overall, these results endorse the superior performance of the new algorithms, which are directly attributable to both the flexibility (i.e., EVPS/VPS selects the best pattern subset, based on the greedy heuristic approach, to represent the RMBs of that video) of the preferential variable pattern selection strategy adopted, and also the new MB classification (Section 2.3). The flexibility is

(a) (b) (c) (d) (e) (f) (g)

Figure 8: (a) *Miss America* frame #2, (b)–(d) Reconstructed frames using the H.264, Fixed-8, and EVPS(8, 75%) algorithms respectively, (e)–(g) Frame differences (×6) of (b), (c), and (d) respectively with respect to (a)

(a) (b) (c) (d) (e) (f) (g)

Figure 9: (a) *Suzie* frame #2, (b)–(d) Reconstructed frames using the H.264, Fixed-8, and EVPS(8, 75%) algorithms respectively, (e)–(g) Frame differences (×6) of (b), (c), and (d) respectively with respect to (a)

To compare the subjective performance, the original frame #2, reconstructed frames, and frame differences are presented in Figure 8 and Figure 9 for the *Miss America* and *Suzie* sequences respectively. Each sequence was coded with 1.5k and 3.1 kbits per frame respectively. The grey-scale intensity of all the absolute frame difference images is magnified by a factor of six to provide a clearer visual comparison. The results for the reconstructed frames using EVPS(8, 75%) can be readily perceived as better than those of the Fixed-8 algorithm and H.264 standard.

6. CONCLUSIONS

Pre-defined fixed pattern set sometimes fail to approximate the moving objects of all video sequences. For better approximation we should increase the pattern set, but a large pattern set requires more bits in pattern identification code for each pattern. So the best solution is to first select the λ best-matched patterns from a pattern codebook of size $\alpha \geq \lambda$ and then representing each moving region with one of the best-matched patterns. In this paper, a novel idea of selecting λ best-matched patterns through preferential selection technique is developed by presenting two algorithms, *Variable Pattern Selection* (VPS) and *Extended VPS* (EVPS) for an initial pattern codebook size of 32. The complexity analysis confirmed that EVPS guaranteed to be nearly six times faster than VPS, with the peak performance providing an improvement factor of nine times. The overall performance of EVPS(8, 75%) was identical to VPS(8) but on average, 0.2 dB and 0.8 dB better than the Fixed-8 algorithm and H.264 standard respectively, for the same number of bits per frame.

Both the VPS and EVPS algorithms rely on a pattern codebook of fixed patterns and hence their efficiency largely depends on the codebook in representing the shape of all kinds of moving regions with minimal overlapping. An inefficient pattern identification numbering scheme can also potentially negate any improvement due to patterns. These algorithms can be improved further by i) generating the pattern codebook from the contents of the video sequence, ii) using different size pattern hierarchy, and iii) using the pattern identification code generated by exploiting correlation among patterns.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the referees and Laurence Dooley for their insightful comments, suggestions and criticisms, which considerably improved the quality of this paper.

REFERENCES

FUKUHARA, T., ASAI, K. and MURAKAMI, T. (1997): Very low bit-rate video coding with block partitioning and adaptive selection of two time-differential frame memories. *IEEE Trans. Circuits Systems Video Technology*, 7: 212–220.

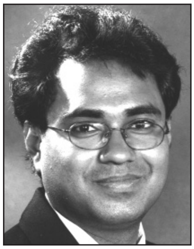
GONZALEZ, R.C. and WOODS, R. E. (1992): *Digital Image Processing*, Addison-Wesley.

ISO/IEC 14496. MPEG-4. (1998): Information Technology-coding of audio-visual objects.

ITU-T Recommendation H.264 (2003): ISO/IEC 14496-10 AVC. Joint Video Team (JVT) of ISO MPEG and ITU-T VCEG, JVT-G050

ITU-T Recommendation H.261. (1993): Video CODEC for audiovisual services at p× 64kbits/s.
ITU-T Recommendation H.263. (1996): Video coding for low bit-rate communication.
ITU-T Recommendation H.263. (1998): Video coding for low bit-rate communication. Version 2.
ITU-T Recommendation H.263. (2000): Video coding for low bit-rate communication. Version 3.
MARAGOS, P. (1987): Tutorial on advances in morphological image processing and analysis. *Opt. Eng.*, 26(7): 623–632.
PAUL, M., MURSHED, M. and DOOLEY, L. (2002): A low bit-rate video-coding algorithm based upon variable pattern selection. *Proc. of 6th IEEE Int. Conf. on Signal Processing* (ICSP-02), Beijing, 2: 933–936.
PAUL, M., MURSHED, M. and DOOLEY, L. (2002): 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* (IKOMAT-02), Crema, Italy, 1560–1564.
PAUL, M., MURSHED, M. and DOOLEY, L. (2002): Impact of macroblock classification on low bit rate video coding focusing on moving region. *Proc. of Int. Conf. of Computer and Information Technology* (ICCIT-2002), Dhaka, Bangladesh, 465–470.
Preferential Voting in Australia, http://www.eca.gov.au/systems .
SHI, Y.Q. and SUN, H. (1999): *Image and Video Compression for Multimedia Engineering Fundamentals, Algorithms, and Standards*, CRC Press.
WONG, K.-W., LAM, K.-M., and SIU, W.-C. (2001): An efficient low bit-rate video-coding algorithm focusing on moving regions, *IEEE trans. circuits and systems for video technology*, 11(10): 1128–1134.

BIOGRAPHICAL NOTES



Manoranjan Paul

Manoranjan Paul received his BScEng (Hons) in Computer Science and Engineering from Bangladesh University of Engineering and Technology (BUET) in 1997. He joined Ahsanullah University of Science and Technology, Bangladesh in 1997 as a lecturer in the Computer Science and Engineering Department, and was promoted to assistant professor in 2000. He was admitted as a PhD student at Gippsland School of Computing and IT (GSCIT) at Monash University, Australia in 2001 and was also appointed as an assistant lecturer (part time) at GSCIT since his PhD candidature. His major research interests are in the fields of image/video coding, multimedia communication, video indexing, video on demand, image segmentation and artificial intelligence. He has published more than 15 refereed international journal, book chapter, and conference publications. Mr Paul is a member of the IEEE and ACS.

Manzur Murshed

Manzur M. Murshed received his BScEng (Hons) degree in Computer Science and Engineering from Bangladesh University of Engineering and Technology (BUET) in 1994 and PhD in Computer Science from the Australian National University in 1999. He is currently the director of research and a senior lecturer at Gippsland School of Computing and Information Technology, Monash University, Australia, where his major research interests are in the fields of multimedia signal processing and communications, parallel and distributed computing, algorithms, simulation, and multilingual systems development. He has published more than 85 journal and peer-reviewed research publications. He is the recipient of numerous academic awards including the University Gold Medal by BUET. He is a member of the IEEE.