

# A Real-Time Pattern Selection Algorithm for Very Low Bit-Rate Video Coding Using Relevance and Similarity Metrics

Manoranjan Paul, *Member, IEEE*, Manzur Murshed, *Member, IEEE*, and Laurence S. Dooley, *Senior Member, IEEE*

**Abstract**—Very low bit-rate video coding using regularly shaped patterns to represent moving regions in macroblocks has good potential for improved coding efficiency. This paper presents a *real-time pattern selection* (RTPS) algorithm, which uses a pattern relevance and similarity metric to achieve faster pattern selection from a large codebook. For each applicable macroblock, the relevance metric is applied to create a *customized pattern codebook* (CPC) from which the best pattern is selected using the similarity metric. The CPC size is adapted to facilitate real-time selection. Results prove the quantitative and perceptual performance of RTPS is superior to both the Fixed-8 algorithm [16] and H.263.

**Index Terms**—Motion compensation, pattern matching, teleconferencing, video coding.

## I. 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 [6]–[8]. These standards are still unable to encode moving objects within a  $16 \times 16$  pixel *macroblock* (MB) during motion estimation, resulting in all 256 residual error values being transmitted for motion compensation regardless of whether there are moving objects or not. One solution is to subdivide the MB and then apply motion estimation and compensation to each subblock. With sufficient numbers of subblocks, the shape of a moving object can be more accurately represented, but this carries a correspondingly higher processing and bit coding overhead [1].

An alternative approach was proposed by Fukuhara *et al.* [1] who used four MB-partitioning patterns each comprising 128-pixels. Motion estimation and compensation were carried out on all eight possible 128-pixel partitions of an MB and the pattern with the lowest prediction error selected. While this gave better performance compared to H.263, not only was the computational complexity of the motion-based processing too high for real-time applications, but also by having only four patterns meant it was insufficient to represent moving objects [16]. By

Manuscript received June 3, 2003; revised March 26, 2004. This paper was recommended by Associate Editor J.-N. Hwang.

M. Paul is with the Gippsland School of Computing and Information Technology, Monash University, Churchill, Vic. 3842, Australia, on leave from Ahsanullah University of Science and Technology, Dhaka 1215, Bangladesh (e-mail: manoranjan@ieee.org).

M. Murshed and L. S. Dooley are with the Gippsland School of Computing and Information Technology, Monash University, Churchill, Vic. 3842, Australia (e-mail: Manzur.Murshed@infotech.monash.edu.au; Laurence.Dooley@infotech.monash.edu.au).

Digital Object Identifier 10.1109/TCSVT.2005.848312

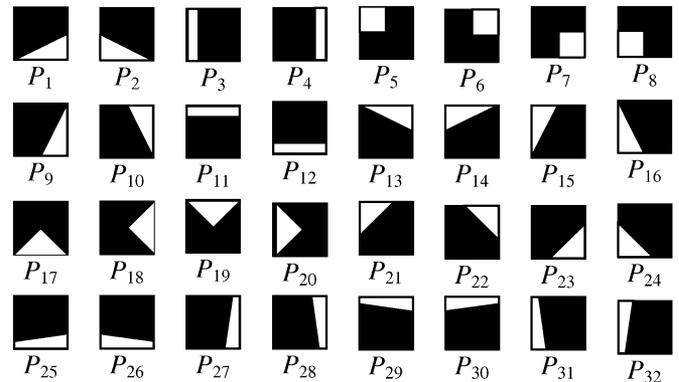


Fig. 1. PC of 32 regular shaped, 64-pixel patterns, defined in  $16 \times 16$  blocks, where the white region represents 1 (motion) and the black region represents 0 (no motion).

treating identically each MB, irrespective of its motion content, also resulted in a higher bit rate being incurred for those MBs which contained only static background or had moving object(s), but with little static background. In such cases, the motion vectors for both partitions were almost the same and so only one could be represented.

The MPEG-4 [5] 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. To address the limitations of Fukuhara's approach [1], Wong *et al.* [16] exploited the idea of partitioning the MBs via a simplified segmentation process that again avoided handling the exact shape of the moving objects, so that popular MB-based motion estimation techniques could be applied. This algorithm focused on the moving regions of the MBs, through the use of a set of regular 64-pixel pattern templates, from a codebook of patterns  $P_1$ – $P_8$  in Fig. 1. If in using some similarity measure, the MR of an 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. Successful pattern matching can therefore, theoretically has a maximum compression ratio of 4:1 for any MB. The actual achievable compression ratio will be lower due to the computing overheads for handling an additional MB type, the pattern identification numbering and pattern matching errors.

Wong *et al.* [16] classified each MB into one of three distinct categories: 1) *static MB* (SMB): MBs that contain little or no motion; 2) *active MB* (AMB): MBs that contain moving object(s) with little static background; and 3) *active-Region MB* (RMB): MBs that contain both static background and part(s)

of moving object(s) covered by one of the pattern in the codebook. The first two MB types are available in the H.263 standard and are treated exactly the same way. For the RMB class, motion estimation and compensation are performed only for those moving regions covered by a selected pattern from the codebook. Overall this provides better prediction and compression efficiency as well as reducing the encoding time compared to H.263 from between 8% and 53% for smooth motion sequences [16].

It was also observed in [16] that the coding efficiency with eight patterns is superior to using only the first four patterns. Throughout this paper, any pattern selection algorithm using the same set of  $\lambda$  patterns for a video sequence is termed as Fixed- $\lambda$  algorithm. The eight-pattern algorithm [16] is, therefore, referred to as the Fixed-8 algorithm. Paul *et al.* [10] and [11] observed a similar trend, but with diminishing returns, when the pattern codebook (PC) size was further extended to 24 and 32 patterns, respectively. The full 32-PC is shown in Fig. 1 where each 64-pixel pattern is *regular*—bounded by straight lines, *clustered*—the pixels are connected, and *boundary-adjointed*. The experimental results presented in this paper will prove that if all 32 codebook patterns are considered for similarity matching for an RMB, on average only 55% are represented by the first eight patterns used in [16].

To counter the diminishing improvement in coding efficiency due to the increased number of bits required to identify each of the 32-patterns, Paul *et al.* [10] developed a *variable pattern selection* (VPS) algorithm to select the  $\lambda$  best-matched pattern set from the codebook using a *greedy* approach, where  $\lambda \in \{4, 8, 16, 24\}$ . Unlike [16], the VPS algorithm has the flexibility of using a different set of  $\lambda$  patterns depending on the moving region variations in the video sequence. The drawback of VPS was that 32- $\lambda$  iterations were required, so an *extended VPS* (EVPS) algorithm was proposed [11], which reduced the computational time while maintaining similar prediction and compression efficiency compared to the VPS algorithm.

The variable pattern selection process required two coding passes for a video sequence. In the first (preprocessing) pass, the  $\lambda$  best-matched pattern set was obtained while in the second (coding) pass, each RMB was matched against one of the pattern from this set using a similarity measure. The computational expense involved in preprocessing however precluded a practical real-time realization of this algorithm. It is important to emphasize that the  $\lambda$  best-matched pattern set is not necessarily the same as the optimal  $\lambda$  patterns that minimize the mean similarity metric for all RMBs. The question of optimality in selecting the subset of patterns from a codebook has recently been addressed by Paul *et al.* [14].

The coding efficiency of these various pattern matching algorithms is largely dependent on the number of RMBs. Wong *et al.* [16] classified an MB as a *candidate RMB* (CRMB) if any of the four  $8 \times 8$  quadrants have no moving pixels present. This quadrant-based classification may in certain instances reduce the number of RMBs by misclassifying a possible CRMB as an AMB because only one or two moving pixels exist in another quadrant. Conversely, the classification may also increase the computational complexity by misclassifying an AMB as a CRMB where all but one quadrant has many moving pixels.

A CRMB is ultimately classified as an RMB depending on a similarity measure with the patterns in the codebook. To overcome these limitations, Paul *et al.* [12] presented a new parametric ( $\delta \in \{64, 96, 128\}$ ) MB classification definition, where the total number of moving pixels in a MB, without considering the quadrants, was compared against  $\delta$  to classify an MB as a CRMB. This technique proved on average, to capture 40% more RMBs than the classification used in the Fixed-8 algorithm, for standard video sequences with  $\delta = 128$ .

Measuring the similarity between a CRMB and all the patterns in the codebook on a piecewise-pixel basis can be very computationally expensive, especially when the codebook size is large, which is always desirable for better coding efficiency. However, it can easily be observed that not all patterns are *relevant* for consideration when using the similarity measure. For example, consider a CRMB whose moving region is best covered by pattern  $P_5$ . For this candidate, all patterns that are not partially covering the moving pixels in  $P_5$ , such as  $P_7$  or  $P_{12}$  may be deemed irrelevant to some degree, depending on their proximity.

In this paper, a gravitational centre proximity-based *pattern relevance* measure is proposed to dynamically create a smaller-sized *customized pattern codebook* (CPC) for each CRMB, by eliminating irrelevant patterns from the original codebook. A new *real time pattern selection* (RTPS) algorithm is then developed to select the best pattern for a CRMB from the CPC, using a piecewise-pixel similarity measure. The rationale in using both relevance and similarity metrics to select the best pattern for a CRMB, is that it provides a facility to trade off between computational complexity and picture quality. In selecting the best pattern, the relevance metric uses only one point (the gravitational centre) to represent all moving pixels in a CRMB, whereas the similarity metric uses all pixels, so there will be an error between the two metrics. However, the relevance metric requires only five add-equivalent operations compared with 767 add-equivalent operations (Section III) for the similarity metric, so it is more than 150 times faster. The RTPS algorithm uses a novel mechanism to control the size of the CPC within predefined bounds, to adapt the computational complexity of the pattern selection process, so ensuring real time operation. RTPS is thus able to process arbitrary-sized codebooks while this real-time constraint is upheld. Furthermore, the computational overhead of the similarity metric is reduced significantly by performing the processing on a quadrant-by-quadrant basis with the option to terminate whenever the measure exceeds a predefined threshold value.

In order to equitably compare the performance of the RTPS algorithm with the Fixed-8 algorithm [16], the average size of the CPC is always kept close to 8. It will be proven that in such circumstances the computational complexity of RTPS is comparable to the Fixed-8 algorithm, while experimental results will reveal that for the same bit rate, the peak signal-to-noise ratio (PSNR) is superior to both the Fixed-8 algorithm and H.263, by up to 0.8 and 1.52 dB, respectively.

This paper is organized as follows. The pattern relevance and similarity metrics as well as the RTPS algorithm along with the MB classification algorithm are detailed in Section II. The actual RTPS coding technique and computational complexity analysis

are presented in Sections III and IV, respectively. Experimental results are fully discussed in Section V to corroborate both the qualitative and quantitative performance of the RTPS algorithm compared with H.263 and the Fixed-8 algorithm. Section VI concludes the paper.

## II. PATTERN RELEVANCE AND SIMILARITY METRICS

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  $\times$   $H$  lines, respectively of a video sequence, where  $0 \leq x, y \leq 15$  and  $0 \leq k < W/16 \times H/16$ . The moving region  $M_k(x, y)$  of the  $k$ th MB in the current frame is obtained as follows:

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

where  $B$  is a  $3 \times 3$  unit matrix for the morphological closing operation  $\cdot$  [2], [9], which is applied to reduce noise, and the thresholding function  $T(v) = 1$  if  $v > 2$  and 0 otherwise.

### A. Pattern Relevance

Let  $G(A)$  be the *gravitational centre* (GC) of a  $16 \times 16$  binary matrix  $A$ , such that

$$G(A) = \left( \frac{\sum_{x=0}^{15} \sum_{y=0}^{15} xA(x, y)}{\sum_{x=0}^{15} \sum_{y=0}^{15} A(x, y)}, \frac{\sum_{x=0}^{15} \sum_{y=0}^{15} yA(x, y)}{\sum_{x=0}^{15} \sum_{y=0}^{15} A(x, y)} \right). \quad (2)$$

For the original PC, the *relevance* of the  $k$ th MB to a pattern  $P_n \in PC$  can be measured as

$$\nabla_{k,n} = \text{dist}(G(M_k), G(P_n)) \quad (3)$$

where  $\text{dist}(a, b)$  is the Manhattan distance between points  $a$  and  $b$ . If the  $k$ th MB is a CRMB then the CPC is formed using the following rule:

$$\forall P_n \in PC : (\nabla_{k,n} \leq T_R) \Rightarrow (P_n \in CPC) \quad (4)$$

where  $T_R$  is the relevance threshold.

It is very important to highlight the role that  $T_R$  has in controlling the size of a CPC, as it provides a low computational complexity filtering mechanism to reduce the pattern set for a particular CRMB prior to the best pattern being selected by the similarity metric. If  $T_R$  is too low, certain CPCs may be an empty set leading inevitably to poorer compression by misclassifying some RBMs as AMBs. Conversely if  $T_R$  is too high, the CPC becomes similarly sized to the full PC, thereby negating the computational benefits of using a small dynamic codebook to facilitate real-time pattern selection.

It is also important to understand that in order to ensure image quality equity to all the CRMBs, the value of  $T_R$  must be kept constant as it directly controls the boundary of proximity under consideration. For the same  $T_R$  value, however, the size of the CPC will vary for different CRMBs. In order to guarantee on the average size of a CPC, instead of setting the value of  $T_R$  arbitrarily, it must be chosen considering the possible minimum

TABLE I  
RELEVANCE THRESHOLD VALUES  $T_R(\eta_{\min})$  FOR A DYNAMICALLY CREATED CPC HAVING A LOWER BOUND OF  $\eta_{\min}$  AND AN UPPER BOUND OF  $\eta_{\max}(\eta_{\min})$  PATTERNS

$\eta_{\min}$	$\eta_{\max}(\eta_{\min})$	$T_R(\eta_{\min})$	$\eta_{\min}$	$\eta_{\max}(\eta_{\min})$	$T_R(\eta_{\min})$
2	11	5.81	15	32	12.72
4	11	6.00	17	32	13.00
6	11	6.19	18	32	13.28
8	17	7.00	20	32	16.38
9	26	8.19	22	32	17.44
10	27	8.44	24	32	18.81
11	31	9.38	26	32	19.00
12	32	10.38	28	32	20.19
13	32	10.81	31	32	21.00
14	32	11.44	32	32	21.44

and maximum sized CPCs. Before explaining this iso- $T_R$  technique in detail, let us consider an obvious way of achieving a tighter control on the size of the CPC irrespective of CRMBs by selecting the first  $j$ th relevant patterns. This straightforward approach, however, not only incurs sorting overhead in addition to the relevance and similarity metrics but also introduces possibility of including patterns (in the CPC) that are too far to be relevant with the corresponding CRMB or missing patterns that are too close to be irrelevant. Experimental results show that the iso- $T_R$  technique based on (4) outperforms this straightforward approach in terms of both image quality and computational complexity for all  $j$ , when the average size of the CPC is matched for both the techniques. The main reason for this improvement is the flexibility of the iso- $T_R$  technique in selecting different numbers of patterns according to the relevance of a CRMB. For example, for the case of  $T_R = 6$  in Table I, between 4 and 11 patterns may be selected. To guarantee an upper and lower bound upon the size of a CPC, the following innovative solution is proposed for the iso- $T_R$  technique.

Suppose  $T_R(\eta_{\min})$  is the minimum value of  $T_R$  that guarantees a lower bound of  $\eta_{\min}$  on the size of any CPC, that is, for a particular CRMB, there will always be at least  $\eta_{\min}$  number of patterns available in its CPC to be tested by the similarity metric. To calculate the exact value of  $T_R(\eta_{\min})$  requires considering all possible real coordinate values as the GCs of potential CRMBs, which is obviously a nondeterministic polynomial (NP)-complete problem. Instead an approximated value is obtained as follows. First, the minimum distance covering at least  $\eta_{\min}$  number of patterns is found by considering each integer coordinate within the MB boundary as the GC of a potential CRMB. The maximum value from these minimum distances is then chosen to guarantee the lower bound irrespective of CRMBs. This approximation technique can be formulated as follows:

$$T_R(\eta_{\min}) = \max_{1 \leq x, y \leq 14} \eta_{\min}^{\text{th}} \min_{\forall P_n \in PC} (\text{dist}((x, y), G(P_n))). \quad (5)$$

Considering only integer coordinates can potentially lead to underestimating the value of  $T_R(\eta_{\min})$ , so reducing the lower bound on the size of certain CPCs. To minimize this likelihood, coordinates on an MB boundary are excluded since this leads

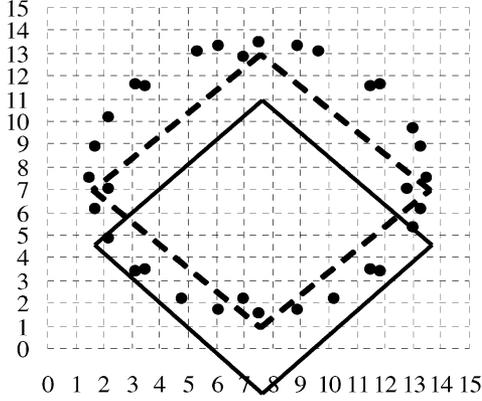


Fig. 2. Example supporting  $T_R(4) = 6$  (dashed diamond) and  $\eta_{\max}(4) = 11$  (solid diamond).

to a tighter bound and in practice, such CRMBs are rarely encountered. To also avoid any ambiguity, all such CRMBs are classified as AMBs.

Interestingly, the computation of  $T_R(\eta_{\min})$  also leads to an upper bound,  $\eta_{\max}(\eta_{\min})$  on the size of a CPC. As with  $T_R(\eta_{\min})$ , because of the infinite number of possible real coordinates, calculating the exact value of  $\eta_{\max}(\eta_{\min})$  is again an NP-complete problem. Instead an approximated upper bound is obtained by taking the maximum size among all the CPCs that can be obtained considering all the integer coordinates within the MB boundary as the GCs of potential CRMBs. This approximation technique can be formulated as follows:

$$\eta_{\max}(\eta_{\min}) = \max_{1 \leq x, y \leq 14} \times \sum_{\forall P_n \in \text{PC}} \begin{cases} 1, & \text{if } \text{dist}((x, y), G(P_n)) \\ & \leq T_R(\eta_{\min}) \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

This approximation may lead to small variations in the number of relevance patterns  $\eta_{\max}(\eta_{\min})$ , but such patterns will always have the least relevance in a CPC so their effect on performance is negligible. Table I presents  $\eta_{\max}(\eta_{\min})$  and  $T_R(\eta_{\min})$  for all possible  $\eta_{\min}$  values. It is observed that there exist some consecutive values of  $\eta_{\min}$  for which the same  $T_R(\eta_{\min})$  is obtained e.g.,  $T_R(3) = T_R(4) = 6$ . In such cases, only the maximum  $\eta_{\min}$  value is displayed. Fig. 2 illustrates how the relevance metric using  $T_R = 6$  can construct a CPC of size as low as 4 and as high as 11, where each dot represents the GC of a pattern. Due to the use of Manhattan distance in (3), the CPC obtained using (4) includes all the patterns in PC for which the GC is covered by the diamond-shaped (a square with sides at  $45^\circ$  with the axes of the coordinate system) area of diagonal value  $2T_R$  with its centre at the GC of the corresponding CRMB. Note that if the Euclidian distance were used, the area would have been a circle of radius  $T_R$  with its centre at the GC of the corresponding CRMB. The dashed and solid diamonds in Fig. 2 cover 4 and 11 patterns, respectively, by considering the two extreme positions.

Using  $T_R = T_R(\eta_{\min})$  in (4) constructs a CPC size between  $\eta_{\min}$  and  $\eta_{\max}(\eta_{\min})$ , with an approximate average of  $\eta_{\text{avg}}(\eta_{\min}) = (\eta_{\min} + \eta_{\max}(\eta_{\min}))/2$ , i.e., the CPC size can

be variable, which is contrary to other pattern selection techniques. To guarantee each CPC has exactly  $\eta_{\text{avg}}(\eta_{\min})$  patterns requires a sorting procedure to identify the best relevant patterns from the codebook. There is, however, no justification for introducing this overhead into RTPS. To appreciate the reason for this, let the corresponding variable and fixed-sized CPCs be  $\text{CPC}_1$  and  $\text{CPC}_2$ , respectively. The following two observations then hold:

- 1) if size of  $\text{CPC}_1 \geq \eta_{\text{avg}}(\eta_{\min})$  then  $\text{CPC}_1 \supseteq \text{CPC}_2$ , which implies the best pattern match from  $\text{CPC}_1$  will be at least as good as that obtained from using  $\text{CPC}_2$ ;
- 2) otherwise, the patterns  $\in (\text{CPC}_2 - \text{CPC}_1)$  are too irrelevant to contribute significantly in improving the coding efficiency.

### B. Pattern Similarity

The *similarity* of the  $k$ th MB to a pattern  $P_n \in \text{CPC}$  can be measured using the following distance:

$$D_{k,n} = \frac{1}{256} \sum_{x=0}^{15} \sum_{y=0}^{15} |M_k(x, y) - P_n(x, y)|. \quad (7)$$

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

$$D_{k,n} = \min_{\forall P_i \in \text{CPC}} (D_{k,i} | D_{k,i} < T_S) \quad (8)$$

where  $T_S$  is the *similarity threshold*. It is assumed in this 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  $\geq 64/256 = 0.25$ .

By exploiting the relational condition in (8), the computational complexity of (7) can be significantly reduced by performing the calculation, in order on a quadrant-by-quadrant level, as

$$D_{k,n}^q = \frac{1}{64} \sum_{x=0}^7 \sum_{y=0}^7 |M_k^q(x, y) - P_n^q(x, y)| \quad (9)$$

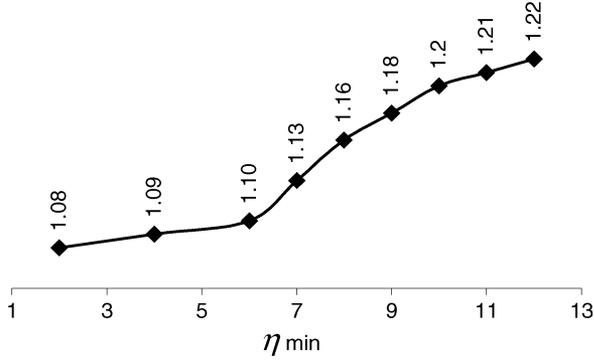
where  $q$  is the quadrant number. The calculation in each quadrant is terminated whenever  $\sum_{i=I}^q D_{k,n}^i \geq T_S$  for  $q = \text{I, II, III, and IV}$ .

Let  $\tau$  be the *speed-up* factor achieved by using this quadrant-based approach compared to the method in [16]. For example, Fig. 3 shows that when  $\eta_{\min} = 6$  for the *Miss America* sequence,  $\tau = 1.1$ , i.e., a 10% saving is attained. As the size of a CPC can only be increased by adding more relatively irrelevant patterns,  $\tau$  increases monotonically with  $\eta_{\min}$  as shown in Fig. 3.

The complete RTPS algorithm is now formally defined in Fig. 4. The  $k$ th MB is then classified into one of the three MB categories using the algorithm in Fig. 5, for all  $k$ .

## III. CODING TECHNIQUES

In this paper, the coding techniques used in [16] are employed with the following exception. Instead of using fixed-length codes, all the 32 patterns in the codebook are identified using the variable-length (Huffman) codes shown in Table II.

Fig. 3. Values of  $\tau$  for different  $\eta_{\min}$  values on the *Miss America* sequence.**Algorithm RTPS( $\eta_{\min}$ , CRMB)****Precondition:**  $PC = \{P_1, \dots, P_{|PC|}\}$ ,  $T_R(\eta_{\min})$ ,  $T_S = 0.25$ .**Return:** BestPattern.**Assumption:** Let CRMB be the  $k^{\text{th}}$  MB.

(Construction of CPC)

1.  $CPC = \emptyset$ ;
2. FOR all  $P_n \in PC$
3. IF  $\forall_{k,n} \leq T_R(\eta_{\min})$  THEN  $CPC = CPC \cup \{P_n\}$ ; ENDIF
4. ENDFOR

(Best Pattern Selection)

5. BestSimilarity =  $\infty$ ; BestPattern = NULL;
6. FOR all  $P_n \in CPC$
7. Sum = 0;
8. FOR  $q = I, II, III, \text{ and } IV$
9. Sum = Sum +  $D_{k,n}^q$ ;
10. IF Sum  $\geq T_S$  THEN GOTO Line 6; ENDIF
11. ENDFOR
12. IF Sum < BestSimilarity THEN
13. BestSimilarity = Sum; BestPattern =  $P_n$ ;
14. ENDFOR
15. ENDFOR

Fig. 4. RTPS algorithm.

**Algorithm CLASSIFY( $k^{\text{th}}$  MB,  $\delta$ ,  $\eta_{\min}$ )****Return:** MBType and BestPattern (if any).

1. IF  $\sum M_k < 8$  THEN
2. MBType = SMB;
3. ELSE
4. IF  $\sum M_k < \delta$  THEN
5. BestPattern = RTPS( $\eta_{\min}$ ,  $k^{\text{th}}$  MB);
6. IF BestPattern is not NULL THEN
7. MBType = RMB;
8. ELSE
9. MBType = AMB;
10. ENDFOR
11. ELSE
12. MBType = AMB;
13. ENDFOR
14. ENDFOR

Fig. 5. MB classification algorithm.

These were obtained from the average pattern frequencies over a large number of standard and nonstandard video sequences.

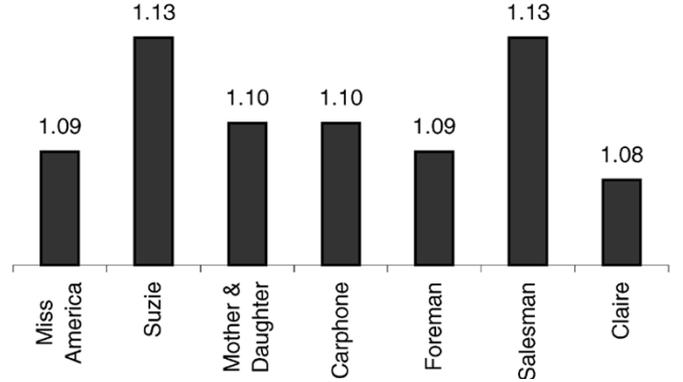
## IV. COMPUTATIONAL COMPLEXITY

## A. Comprehensive Analysis

**Lemma 1:** The construction of a CPC from an original PC in the RTPS algorithm requires on average  $\delta + 6|PC| + 261$  “add-equivalent” and 2 “division” operations.

TABLE II  
PATTERN ID NUMBER  $\Rightarrow$  VARIABLE LENGTH CODE

1 $\Rightarrow$ 1010	9 $\Rightarrow$ 010011	17 $\Rightarrow$ 001100	25 $\Rightarrow$ 111011
2 $\Rightarrow$ 00101	10 $\Rightarrow$ 11100	18 $\Rightarrow$ 010010	26 $\Rightarrow$ 0011101
3 $\Rightarrow$ 00001	11 $\Rightarrow$ 01000	19 $\Rightarrow$ 001101	27 $\Rightarrow$ 0101011
4 $\Rightarrow$ 00000	12 $\Rightarrow$ 01011	20 $\Rightarrow$ 001111	28 $\Rightarrow$ 0010000
5 $\Rightarrow$ 011	13 $\Rightarrow$ 001001	21 $\Rightarrow$ 0010001	29 $\Rightarrow$ 111010
6 $\Rightarrow$ 100	14 $\Rightarrow$ 10110	22 $\Rightarrow$ 101111	30 $\Rightarrow$ 11100
7 $\Rightarrow$ 110	15 $\Rightarrow$ 101110	23 $\Rightarrow$ 010100	31 $\Rightarrow$ 0011100
8 $\Rightarrow$ 0001	16 $\Rightarrow$ 11111	24 $\Rightarrow$ 111101	32 $\Rightarrow$ 0101010

Fig. 6. Speed-up factor  $\tau$  for  $\eta_{\min} = 4$  on seven standard sequences.

**Proof:** A CRMB will have a minimum of 8 and a maximum of  $\delta - 1$  moving pixels. Calculating the gravitational centre of a CRMB using (2) requires on average: 256 “compare”,  $\delta + 5$  “add,” and 2 “division” operations. The gravitational centre of every pattern is known *a priori*. Calculating the Manhattan distance of this centre with the centres of all the patterns in the PC requires  $2|PC|$  “subtract,”  $|PC|$  “add,” and  $2|PC|$  “absolute” operations. One “comparison” operation is required (Line 3) in the RTPS algorithm. ■

**Lemma 2:** The best pattern selection in the RTPS algorithm requires on average  $769 \times \eta_{\text{avg}}(\eta_{\min})/\tau$  “add-equivalent” operations.

**Proof:** The similarity measure in (7) requires 255 “add,” 256 “absolute,” 256 “subtract,” and 1 “shift” operations. One “comparison” operation is required (Line 10) in the RTPS algorithm. A quadrant-by-quadrant level similarity metric, therefore, requires  $769/\tau$  “add-equivalent” operations. The average  $|CPC|$  size is  $\eta_{\text{avg}}(\eta_{\min})$ . ■

As the Fixed-8 algorithm does not apply any relevance metric, the following lemma can be proven by means of a similar argument to that used in Lemma 2:

**Lemma 3:** The Fixed-8 algorithm requires  $769 \times 8 = 6152$  “add-equivalent” operations. ■

**Theorem 1:** The RTPS(4) algorithm using  $\delta = 128$  and  $|PC| = 32$  has the same computational complexity as the Fixed-8 algorithm.

**Proof:** From Table I,  $\eta_{\text{avg}}(4) = (4 + 11)/2 = 7.5$ , while in Fig. 6, the average speed-up factor for  $\eta_{\min} = 4$  is  $\tau = 1.1$ . Using Lemmas 1 and 2, the RTPS(4) algorithm with  $\delta = 128$  and  $|PC| = 32$  requires  $128 + 6 \times 32 + 261 + 769 \times 7.5/1.1 = 5825$  “add-equivalent” and 2 “division” operations, which represents 5.3% fewer operations than for the Fixed-8 algorithm (Lemma 3). While  $\eta_{\text{avg}}(4)$  can be greater than 7.5

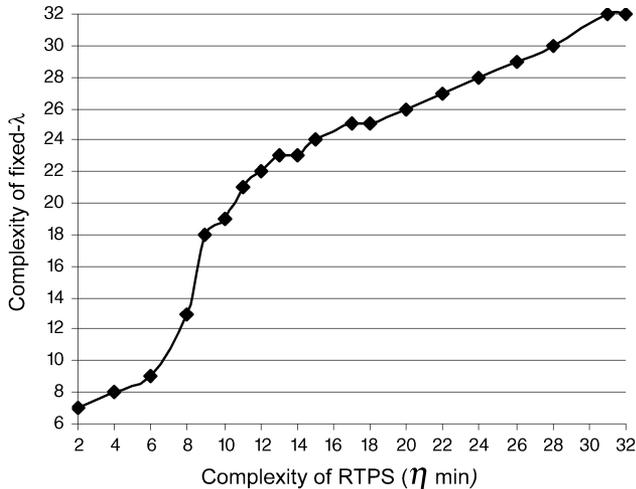


Fig. 7. Comparison of the complexity of the RTPS( $\eta_{\min}$ ) and Fixed- $\lambda$  algorithms.

for some video sequences, the number of operations required for the RTPS(4) algorithm will never be greater than that in the Fixed-8 algorithm provided  $\eta_{\text{avg}}(4) \leq 8$ . This upper bound can be further increased by exploiting the order of the quadrant level similarity calculations using information relating to the GC of the CRMB, which leads to a higher  $\tau$  value. It can, therefore, be concluded that the computational efficiency of the RTPS(4) algorithm is equivalent to that of the Fixed-8 algorithm. ■

### B. Intuitive Approach

The computational overhead involved in the similarity metric is always greater than that of the relevance metric. While a detail mathematical analysis of the number of operations involved is omitted, the following intuitive and simplified conclusion can be made from Lemmas 1 and 2:

*Theorem 2:* The computational complexity of a pattern selection algorithm is directly proportional to the average (integer) number of patterns used in the similarity metric. ■

Fig. 7 provides a comparison of the computational overhead of the RTPS( $\eta_{\min}$ ) versus Fixed- $\lambda$  algorithms using Theorem 2. While the graph supports Theorem 1, it clearly demonstrates the benefit of the RTPS algorithm in being able to control the computational complexity across a wide range with only one degree of freedom (parameter  $\eta_{\min}$ ). Crucially however, in doing so, the RTPS algorithm always makes use of the entire codebook at some stage of the selection process. RTPS can thus support real-time pattern selection for each RMB by considering, the maximum number of pattern similarity measures that are able to be supported by the hardware concerned. Moreover, the basic principles used in the RTPS algorithm can be easily extended to arbitrarily sized PCs.

## V. EXPERIMENTAL RESULTS

Both the RTPS and Fixed-8 algorithms were tested on a large number of standard and nonstandard video sequences of QCIF digital video format [15] containing different degrees of object and camera motion. For brevity all the experimental results are presented using the first 100 frames of seven popular test

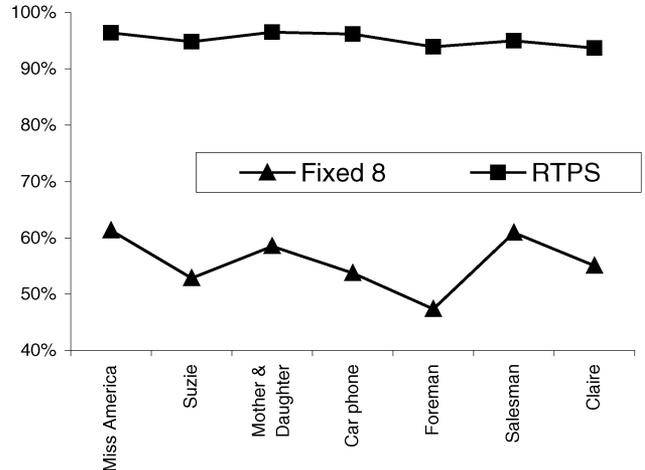


Fig. 8. Percentage of RBMs where the same pattern is selected by the RTPS(4) and Fixed-32 algorithms and the Fixed-8 and Fixed-32 algorithms.

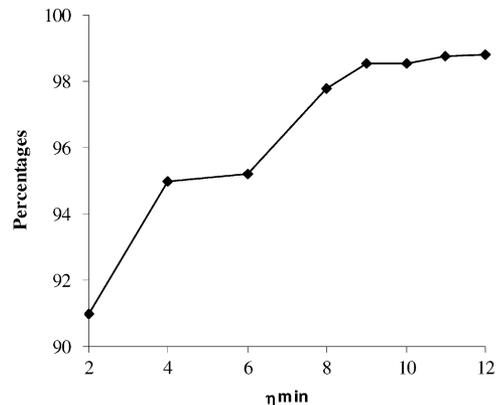


Fig. 9. Percentage of RBMs in the *Miss America* sequence where the same pattern is selected by both the RTPS( $\eta_{\min}$ ) and Fixed-32 algorithms.

video sequences. The motion estimation used a full-search block matching algorithm with half-pel [15] accuracy. Although, the performance of the Fixed-8 algorithm has already been demonstrated better than that of the H.263 standard [16], the latter is included for comparative purposes.

### A. Comparison With the Optimal Fixed-32 Algorithm

For the PC in Fig. 1, the Fixed-32 algorithm always selects the *optimal* pattern for each RMB. This means that all other pattern selection algorithms that use a subset of patterns from this codebook in the similarity measure, e.g., the RTPS( $\eta_{\min}$ ) algorithm,  $\eta_{\min} \leq 11$ , and the Fixed- $\lambda$  algorithm,  $\lambda < 32$ , can at best only match the optimal result obtained using the Fixed-32 algorithm.

Fig. 8 shows that when the RTPS(4) algorithm selected the optimal pattern for 95% of the RBMs, the Fixed-8 algorithm was only able to select on average 55% of the RBMs in the test video sequences. This observation not only reaffirms the significance of extending the size of PC from 8 to 32 as originally proposed in [11] but also reveals a key benefit of the RTPS algorithm, which is that it is able to select the optimal pattern in a very high number of cases, while using a pattern similarity measure of only around eight patterns. Fig. 9 further demonstrates that for the *Miss America* sequence where the RTPS al-

TABLE III  
PERCENTAGE OF DIFFERENT MB TYPES GENERATED BY THE RTPS(11),  
RTPS(4), AND FIXED-8 ALGORITHMS

Video sequences	RTPS(11)			RTPS(4)		Fixed-8	
	SMB	RMB	AMB	RMB	RMB	RMB	RMB
MissAmerica	62	21	16	21			15
Suzie	29	25	46	24			14
Mother & Daughter	43	29	28	28			17
Carphone	22	28	50	28			18
Foreman	14	28	58	27			16
Salesman	57	29	14	28			20
Claire	78	14	8	14			11

TABLE IV  
PSNR OF STANDARD SEQUENCES USING THE H.263, FIXED-8, AND  
RTPS(4) ALGORITHMS

Video sequences	Bit-rate (kbps)	PSNR (dB)		
		H.263	Fixed-8	RTPS(4) with $\delta = 128$
Miss America	23.67	36.10	36.42	37.11
Suzie	42.89	31.66	31.81	32.38
Mother & Daughter	35.72	31.02	31.39	31.70
Car phone	48.63	30.55	30.74	31.06
Foreman	61.79	28.57	28.50	28.78
Salesman	30.88	30.16	30.29	30.61
Claire	18.62	33.70	34.41	35.22

gorithm selected the optimal pattern for more than 99% cases using a  $\eta_{\min}$  value as low as 9. Moreover, it has been empirically proven that the RTPS(11) algorithm performs as well as the optimal Fixed-32 algorithm, while requiring  $(32 - 21)/32 = 35\%$  fewer operations (Fig. 7).

Table III shows the percentage of SMB, RMB, and AMB for selected standard test sequences. The relationship between the number of AMBs and overall bit rate is clearly evident in the table where the larger the number of AMBs, the higher the bit rate. RTPS(4) provides superior performance to the Fixed-8 algorithm because it captures additional RMBs by classifying more CRMBs into RMBs, so reducing the number of AMBs.

### B. Objective Quality Assessment

For the same bit rate, the RTPS(4) algorithm consistently outperforms H.263 and the Fixed-8 algorithm in terms of achieving a higher PSNR, in comparing the reconstructed frames with the original, for all test video sequences. Table IV presents the comparative average PSNR values for the first 100 frames of the seven standard sequences. The RTPS(4) algorithm improved PSNR in the range of 0.28–0.81 dB from the Fixed-8 algorithm and between 0.21 and 1.52 dB against H.263 for the standard test sequences shown in Table IV.

It is especially noteworthy that while the RTPS(4) algorithm improved the PSNR of the *Foreman* sequence by only 0.21 dB, the Fixed-8 algorithm actually degraded the PSNR value by 0.07 dB, an anomaly that was reported in [16]. The main reasons for the RTPS(4) algorithm consistently outperforming H.263 are: 1) the extended MB classification definition [12] and 2) the use of an enlarged PC. The RTPS(4) algorithm improved PSNR

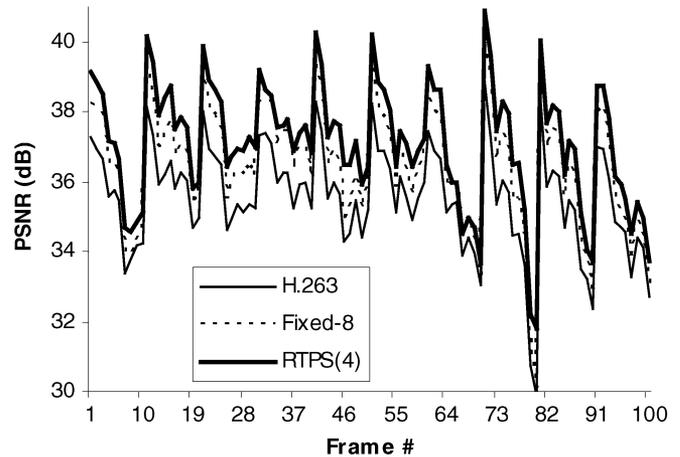


Fig. 10. Frame level PSNR in the *Miss America* sequence using the H.263, Fixed-8, and RTPS(4) algorithms for target bit rate 23.67 kb/s.

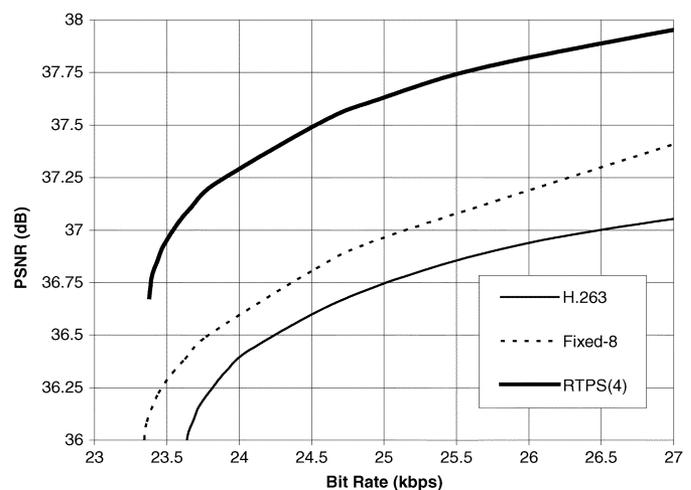


Fig. 11. Average PSNR for the *Miss America* sequence using the H.263, Fixed-8, and RTPS(4) algorithms at different bit rates.

consistently even at the frame level as evidenced in Fig. 10 for the *Miss America* sequence.

Fig. 11 further shows that the PSNR improvement of the RTPS(4) algorithm is consistent across different operating bit rates. In the *Miss America* example, RTPS(4) improved the average PSNR by 1 and 0.75 dB, respectively, compared with H.263 and the Fixed-8 algorithm, for the full range of operating bit rates between 23.5–27 kb/s.

### C. Subjective Quality Assessment

The human visual system does not respond to stimuli in a straightforward manner. It is therefore, widely accepted that objective assessment based on PSNR does not always provide reliable assessments of image quality, since a higher PSNR may not always guarantee better image quality [15]. It has become common practice in international coding-standard activities to combine both objective and subjective assessments in evaluating and comparing video coding algorithms.

To compare the perceptual performance of the three relevant algorithms, the original frame #3, reconstructed frames, and

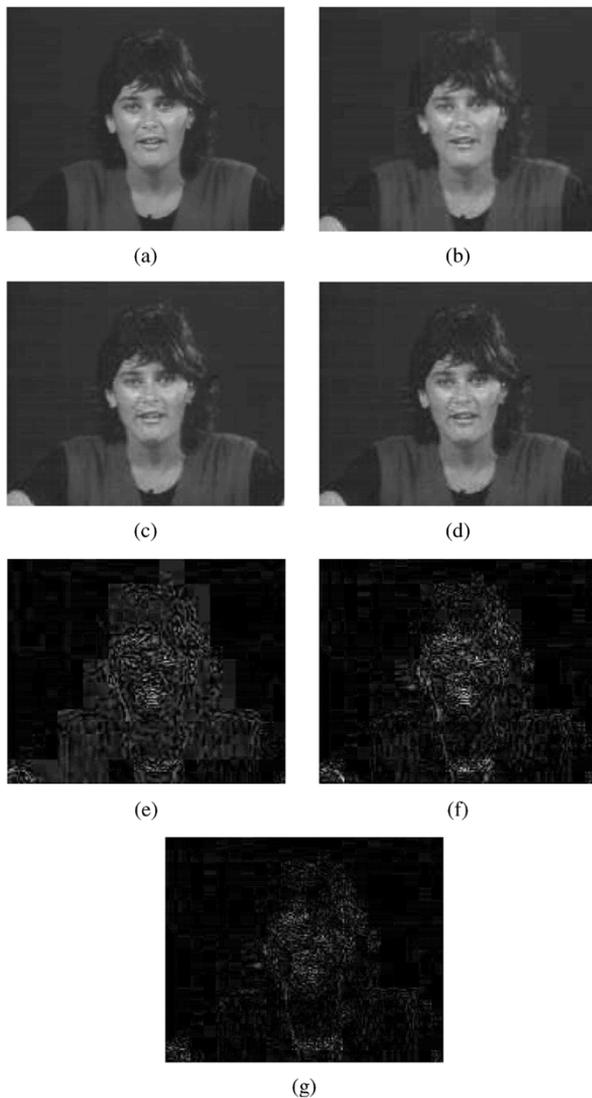


Fig. 12. (a) *Miss America* frame 3. (b)–(d) Reconstructed frames using the H.263, Fixed-8, and RTPS(4) algorithms, respectively. (e)–(g) Frame differences ( $\times 3$ ) of (b), (c), and (d), respectively with respect to (a).

frame differences are presented in Figs. 12 and 13 for the *Miss America* and *Claire* sequences, respectively. The particular bit rates used in coding these two sequences for all three algorithms are 23.67 kb/s for *Miss America* and 18.62 kb/s for *Claire*. The intensity of each frame difference image has been magnified by a factor of three in order to provide an improved visual comparison. In both examples, reconstructed frames using the RTPS(4) algorithm can be readily perceived as superior to those of the Fixed-8 and H.263 algorithms, so endorsing the enhanced quantitative performance of the RTPS algorithm that was highlighted in previous sections.

## VI. CONCLUSION

This paper has presented a new RTPS algorithm which innovatively incorporates both a pattern relevance and similarity metric to achieve faster pattern selection from an original 32-PC. A novel strategy for dynamically controlling the size of a CPC has been developed with upper and lower bounds

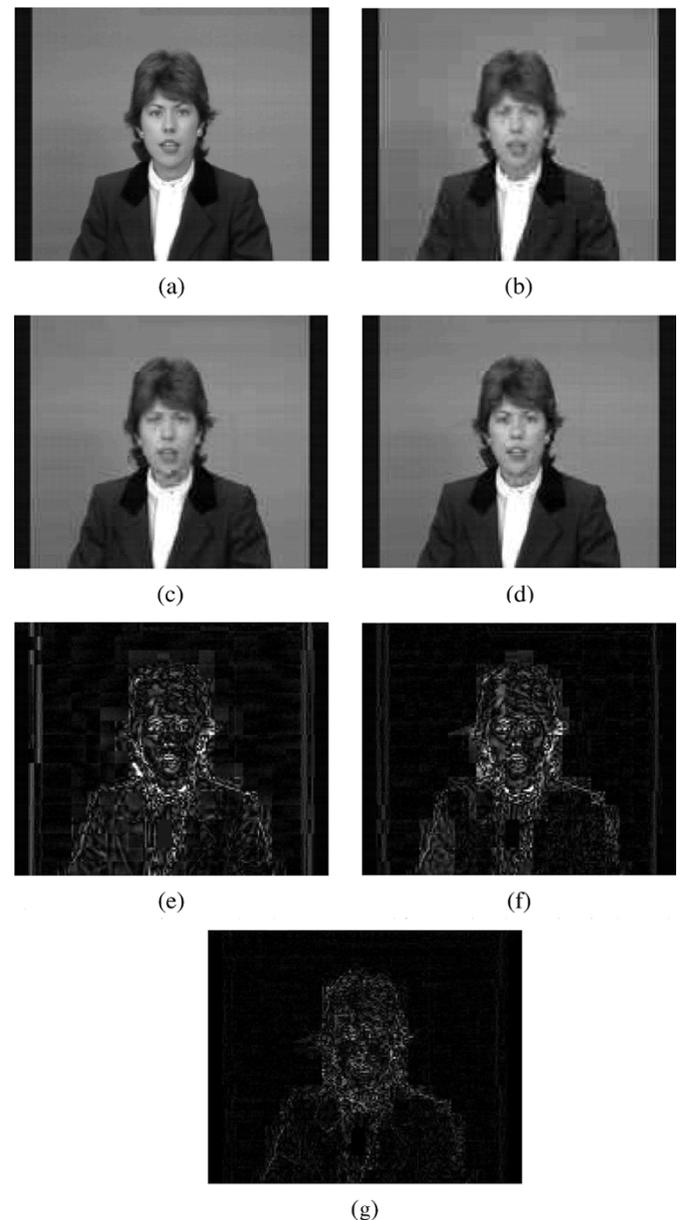


Fig. 13. (a) *Claire* frame 3. (b)–(d) Reconstructed frames using the H.263, Fixed-8, and RTPS(4) algorithms, respectively. (e)–(g) Frame differences ( $\times 3$ ) of (b), (c), and (d), respectively with respect to (a).

defined. RTPS algorithm can control the computational complexity across a wide range by conditioning this lower bound. This arrangement ensures the RTPS algorithm always uses the complete codebook at some stage of the pattern selection process and still manages to keep the computational complexity within real-time constraint. This principal can be easily extended to arbitrarily sized PCs. The computational efficiency of the similarity measure is significantly improved by using a predefined threshold and computing the metric on a quadrant-by-quadrant basis. Overall, the computational efficiency for the RTPS(4) algorithm has been proven to be commensurate with the Fixed-8 algorithm, while for the same bit rate, the quantitative and qualitative performance of the RTPS(4) algorithm is superior to both the Fixed-8 algorithm and H.263 low bit rate video coding standard. RTPS(4) improved the

PSNR value for all experimental test sequences by up to 0.81 dB compared with the Fixed-8 algorithm and up to 1.52 dB for H.263.

#### ACKNOWLEDGMENT

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

#### REFERENCES

- [1] T. Fukuhara, 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.*, vol. 7, no. 1, pp. 212–220, Feb. 1997.
- [2] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Reading, MA: Addison-Wesley, 1992.
- [3] *International Standard*, ISO/IEC 11 172, 1992.
- [4] *MPEG-2 International Standard*, ISO/IEC 13 818, 1995.
- [5] *MPEG-4 International Standard*, ISO/IEC N4030, 2001.
- [6] *Video Codec for Audio-Visual Services at  $p \times 64$  kbits/s*, ITU-T Recommendation H.261, 1993.
- [7] *Video Coding for Low Bit-Rate Communication*, ITU-T Recommendation H.263, 1996.
- [8] *Video Coding for Low Bit-Rate Communication*, ITU-T Recommendation H.263, 1998.
- [9] P. Maragos, "Tutorial on advances in morphological image processing and analysis," *Opt. Eng.*, vol. 26, no. 7, pp. 623–632, 1987.
- [10] M. Paul, M. Murshed, and L. Dooley, "A low bit-rate video-coding algorithm based upon variable pattern selection," in *Proc. 6th Int. Conf. Signal Processing (ICSP-02)*, vol. 2, Beijing, China, 2002, pp. 933–936.
- [11] M. Paul, 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," in *Proc. Int. Workshop Knowledge Management Technique (IKOMAT-02)*, Crema, Italy, 2002, pp. 1560–1564.
- [12] —, "Impact of macroblock classification on low bit rate video coding focusing on moving region," in *Proc. Int. Conf. Computer and Information Technology (ICCIT-2002)*, Dhaka, Bangladesh, 2002, pp. 465–470.
- [13] —, "A new real-time pattern selection algorithm for very low bit-rate video coding focusing on moving regions," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP-03)*, vol. 3, Hong Kong, 2003, pp. 397–400.
- [14] —, "A real time generic variable pattern selection algorithm for very low bit-rate video coding," in *Proc. IEEE Int. Conf. Image Processing (ICIP-03)*, vol. 3, Barcelona, Spain, 2003, pp. 857–860.
- [15] Y. Q. Shi and H. Sun, *Image and Video Compression for Multimedia Engineering Fundamentals, Algorithms, and Standards*. Boca Raton, FL: CRC, 1999.
- [16] K.-W. Wong, K.-M. Lam, and W.-C. Siu, "An efficient low bit-rate video-coding algorithm focusing on moving regions," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 10, pp. 1128–1134, Oct. 2001.



**Manoranjan Paul** (M'03) received the B.Sc.Eng. (hons.) degree in computer science and engineering from Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh, in 1997.

He joined the Computer Science and Engineering Department, Ahsanullah University of Science and Technology, Dhaka, Bangladesh as a Lecturer in 1997 and was promoted to Assistant Professor in 2000. He is currently working toward the Ph.D. degree at Gippsland School of Computing and IT (GSCIT), Monash University, Churchill, Australia,

since 2001 and was also appointed as an Assistant Lecturer (part time). 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 chapters, and conference publications.

Mr. Paul is a Member of the Australian Computer Society (ACS).



**Manzur M. Murshed** (M'96) received the B.Sc.Eng. (hons.) degree in computer science and engineering from Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh, in 1994 and the Ph.D. degree in computer science from the Australian National University, Canberra, Australia, in 1999.

He is currently the Director of Research and a Senior Lecturer at Gippsland School of Computing and Information Technology, Monash University, Churchill, Australia, where his major research

interests are in the fields of video coding and transcoding, video indexing and retrieval, video-on-demand, image processing, multimedia communications, wireless communications, parallel, and distributed computing, grid computing, simulation, complexity analysis, multilingual systems, algorithms, digital watermarking, and distributed coding. He has published more than 70 journal and peer-reviewed research publications.

Dr. Murshed is the recipient of numerous academic awards including the University Gold Medal from BUET.



**Laurence S. Dooley** (M'81-SM'93) received the B.Sc. (hons), M.Sc., and Ph.D. degrees in electrical and electronic engineering from the University of Wales, Swansea, U.K., in 1981, 1983, and 1987, respectively.

Since 1999, he has been Professor of Multimedia Technology in the Gippsland School of Computing and Information Technology, Monash University, Churchill, Australia, where his major recent research interests are in multimedia signal processing, shape-based video object coding, mobile communications, bioinformatics, and smart-sensor networks. He has published over 100 international scientific peer-reviewed journals, book chapters, and conference papers, as well as serving on numerous international technical program committees. He is also currently Executive Director of the Monash Regional Centre for Information and Communications Technology, which has established a regionally based technology transfer gateway network (TTGN) to stimulate the innovation cycle of small business, and provide a bridge for commercialization of new technologies for its industry partners. The TTGN project is partly funded by the Commonwealth AusIndustry *Innovation Access* program.

Prof. Dooley is a Chartered Engineer (C.Eng.) and a corporate member of the British Computer Society (MBCS).