## RESEARCH

Open Access

# Video coding using arbitrarily shaped block partitions in globally optimal perspective

Manoranjan Paul[1*] and Manzur Murshed[2]

**Abstract**

Algorithms using content-based patterns to segment moving regions at the *macroblock* (MB) level have exhibited good potential for improved coding efficiency when embedded into the H.264 standard as an extra mode. The *content-based pattern generation* (CPG) algorithm provides local optimal result as only one pattern can be optimally generated from a given set of moving regions. But, it failed to provide optimal results for multiple patterns from entire sets. Obviously, a global optimal solution for clustering the set and then generation of multiple patterns enhances the performance farther. But a global optimal solution is not achievable due to the non-polynomial nature of the clustering problem. In this paper, we propose a near-*optimal content-based pattern generation* (OCPG) algorithm which outperforms the existing approach. Coupling OCPG, generating a set of patterns after clustering the MBs into several disjoint sets, with a direct pattern selection algorithm by allowing all the MBs in multiple pattern modes outperforms the existing pattern-based coding when embedded into the H.264.

**Keywords:** video coding, block partitioning, H.264, motion estimation, low bit-rate coding, occlusion

## 1. Introduction

VIDEO coding standards such as H.263 [1] and MPEG-2 [2] introduced block-based *motion estimation* (ME) and *motion compensation* (MC) to improve coding performance by capturing various motions in a small area (for example, a $8 \times 8$ block). However, they are inefficient while coding at low bit rate due to their inability to exploit *intra-block temporal redundancy* (ITR). Figure 1 shows that objects can partly cover a block, leaving highly redundant information in successive frames as background is almost static in co-located blocks. Inability to exploit ITR results in the entire $16 \times 16$-pixel *macroblock* (MB) being coded with ME&MC regardless of whether there are moving objects in the MB.

The latest video coding standard H.264 [3] has introduced tree-structured variable block size ME & MC from $16 \times 16$-pixel down to $4 \times 4$-pixel to approximate various motions more accurately within a MB. We empirically observed in [4] that while coding head-and-shoulder type video sequences at low bit rate, more than 70% of the MBs were never partitioned into

smaller blocks by the H.264 that would otherwise be at a high bit-rate. In [5], it has been further demonstrated that the partitioning actually depends upon the extent of motion and *quantization parameter* (QP): for low motion video, 67% (with low QP) to 85% (with high QP) of MBs are not further partitioned; for high motion video, the range is 26-64. It can be easily observed that the possibility of choosing smaller block sizes diminishes as the target bit-rate is lowered. Consequently, coding efficiency improvement due to the variable blocks can no longer be realized for a low bit rate as larger blocks have to be chosen in most cases to keep the bit-rate in check but at the expense of inferior shape and motion approximation.

Recently, many researchers [6-12] have successfully introduced other forms of block partitioning to approximate the shape of a *moving region* more accurately to improve the compression efficiency. Chen et al. [6] extended the variable block size ME&MC method to include additional four partitions each with one L-shaped and one square segment to achieve improvement in picture quality. One of the limitations of segmenting MBs with rectangular/square shape building blocks as done in the method with variable block size and in [6] is that the partitioning boundaries cannot always

* Correspondence: manoranjan@ieee.org
[1]School of Computing and Mathematics, Charles Sturt University, Panorama Avenue, Bathurst, NSW 2795, Australia
Full list of author information is available at the end of the article

approximate arbitrary shapes of moving objects efficiently.

Hung et al. [7] and Divorra et al. [8,9] independently addressed this limitation with the variable block size ME&MC by introducing additional wedge-like partitions where a MB is segmented using a straight line modelled by two parameters: orientation angle $\theta$ and distance $\rho$ from the centre of the MB. A very limited case with only four partitions ($\theta \in \{0°, 45°, 90°, 135°\}$ and $\rho = 0$) was reported by Fukuhara et al. [10] even before the introduction of variable block size ME&MC for low bit rate video coding. Chen et al. [11] and Kim et al. [12] improved compression efficiency further with *implicit block segmentation* (IBS) and thus avoided explicit encoding of the segmentation information. In both cases, the segmentation of the current MB can be generated by the encoder and decoder using previously coded frames only.

But none of these techniques, including the H.264 standard, allows for encoding a block-partitioned segment by skipping ME&MC. Consequently, they use unnecessary bits to encode almost zero motion vectors with perceptually insignificant residual errors for the background segment. These bits are quite valuable at low bit rate that could otherwise be spent wisely for encoding residual errors in perceptually significant segments. Note that the H.264 standard acknowledges the penalty of extra bits used by the motion vectors by imposing rate-distortion optimisation in motion search to keep the length of the motion vector smaller and disallowing B-frames which require two motion vectors, in the *Baseline* profile used widely in video conferencing and mobile applications.
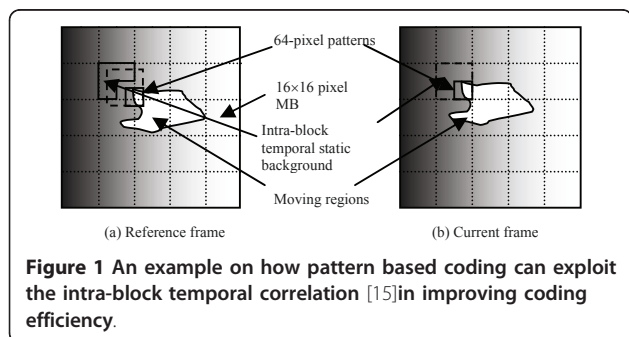
Pattern-based video coding (PVC) initially proposed by Wong et al. [13] and later extended by Paul et al. [14,15] used 8 and 32 pre-defined regular-shaped binary rectangular and non-rectangular pattern templates respectively to segment the moving region in a MB to exploit the ITR. Note that a pattern template is a size of $16 \times 16$ positions (i.e., similar to a MB size) with 64 '1's and 192 '0's. The best-matched moving region of a MB with a pattern template (see in Figure 1) through

an efficient similarity measure estimates the motion, compensates the residual error using only pattern covered-region (i.e., only 64 pixels among 256 pixels), and ignores the remaining region (which is copied from the reference block) of the MB from signalling any bits for motion vector and residual errors. Successful pattern matching can, therefore, theoretically attain maximum compression ratio of 4:1 for a MB as the size of the pattern is 64-pixel. The actual compression however will be lower due to the overheads of identifying this special type of MB as well as the best matched pattern for it and the matching error for approximating the moving region using the pattern. An example of pattern approximation using pre-defined thirty two patterns [14] for *Miss America* video sequence is shown in Figure 2.

As the objects in video sequences are widely varied, not necessarily the moving region is well-matched with any predefined regular-shape pattern template. Intuitively, an efficient coding is possible if the moving region is encoded using the pattern templates generated from the content of the video sequences. Very recently, Paul and Murshed [16] proposed a *content-based pattern generation* (CPG) algorithm to generate eight patterns from the given moving regions. The PVC using those generated patterns outperformed the H.264 (i.e., baseline profile) and the existing PVC by 1.0 and 0.5 dB respectively [16] for head-shoulder-type video sequences. They also mathematically proved that this pattern generation technique is optimal if only one pattern would be generated for a given division of moving regions. Thus, they got a local optimal solution as they could generate single pattern rather than multiple patterns. But for efficient coding, multiple patterns are necessary for different shape of moving regions.

It is obvious that a global optimal solution improves the pattern generation process for multiple patterns, and hence, eventually the coding efficiency. A global optimal solution can be achieved if we are able to divide the entire moving regions optimally. But, this problem is a *non polynomial* (NP-complete) problem, as no clustering techniques provide optimal clusters. In this paper, we propose a heuristic to find the near-optimal clusters and apply local optimal CPG algorithm on each cluster to get the near global optimal solution.

Moreover, the existing PVC used a pre-defined threshold to reduce the number of MBs coded using patterns to control the computational complexity as it requires extra ME cost. It is experimentally observed that any fixed threshold for different video sequences may overlook some potential MBs from the pattern mode [15]. Obviously, eliminating this threshold by allowing all MBs to be motion estimated and



(a) Reference frame          (b) Current frame

**Figure 1 An example on how pattern based coding can exploit the intra-block temporal correlation** [15]**in improving coding efficiency**.
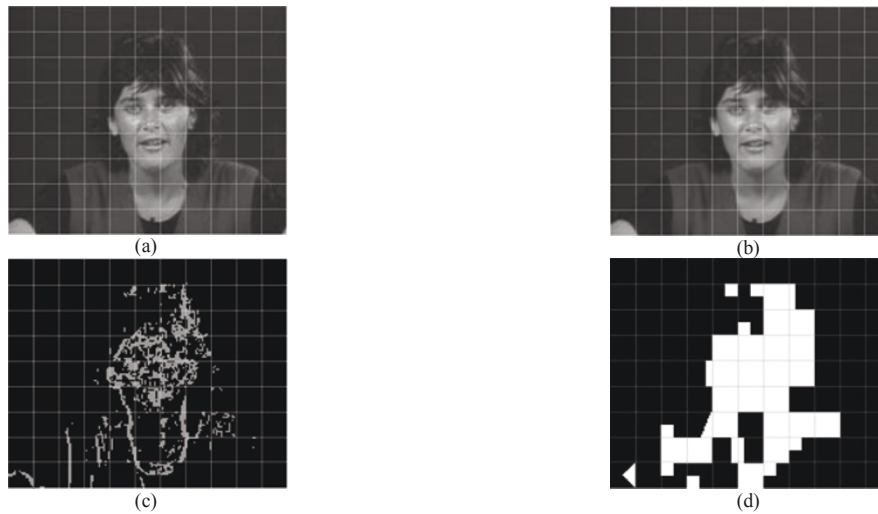
**Figure 2 An example of pattern approximation for the *Miss America* standard video sequence, (a) frame number one, (b) frame number two, (c) detected moving regions, and (d) results of pattern approximation**.

compensated using patterns and finally selected by the Lagrangian optimization function will provide better rate-distortion performance by increasing computational time. To reduce the computational complexity we assume the already known motion vector of the H.264 in pattern mode, which may degrade the performance. But the net performance gain would outweigh this.

As the best pattern selection process solely relies on the similarity measures, it is not guaranteed that the best pattern will always result in maximum compression and better quality, which also depends on the residual errors after quantization and Lagrangian multiplier. This paper also exploits to introduce additional pattern modes that select the pattern in order of similarity ranking. Furthermore, a new Lagrangian multiplier is also determined as the pattern modes provide relatively less bits and slightly higher distortion as compared to the other modes of the H.264. The experimental results confirm that this new scheme successfully improves the rate-distortion performance as compared to the existing PVC as well as the H.264.

The rest of the paper is organized as follows: Section 2 provides the background of the content-based PVC techniques including collection of moving regions & generation of pattern templates, and encoding & decoding of PVC using content-based patterns. Section 3 illustrates the proposed approach including optimal pattern generation technique and its parameter settings. Section 4 discusses the computational complexity of the proposed technique. Section 5 presents the experimental set up along with the comparative performance results. Section 6 concludes the paper.

## 2. Content-based PVC algorithm

The PVC with a set of content-based patterns termed as pattern codebook performs in two phases. In first phase, moving regions are collected from the given number of frames and pattern codebook is generated from those MRs using the CPG algorithm [15]. In the second phase, actual coding is taken place using the generated pattern codebook.

### 2.1. Collection of moving regions and generation of pattern codebook

The moving region in a current MB is defined based on the number of pixels whose intensities are different from the corresponding pixels of the reference MB. The moving region $M$ of a MB $\Omega$ in the current frame is obtained using the co-located MB $\omega$ in the reference frame [13] as follows:

$$M(x, \gamma) = T(|\Omega(x, \gamma) \bullet \Theta - \omega(x, \gamma) \bullet \Theta|), \quad 0 \le x, \gamma \le 15 \quad (1)$$

where $\Theta$ is a $3 \times 3$ unit matrix for the morphological closing operation denoted by $\bullet$ [17], which is applied to reduce noise, and the thresholding function $T(\nu) = 1$ if $\nu > 2$ (i.e., the said pixel intensity difference is bigger than two grey levels) and 0 otherwise. Let $|M|_1$ be the total number of 1's in the matrix $M$. If $8 \le |M|_1 < 2QP/3 + 64$ where $QP$ is the quantization parameter, the corresponding MB, i.e., $\Omega$ will participate in the pattern generation process as it has a reasonable number of moving pixels to be covered by a 64-pixel '1's in a pattern so that high matching error is avoided.

The binary moving region map of $\Omega$ is used in the pattern generation process as the representative of $\Omega$.

The MB with moving region is named as *candidate region-active MB* (CRMB). We have assumed that if the number of '1's in a CRMB is too low or too high, the corresponding MB is not suitable to be encoded by the pattern mode, and thus, we do not include these CRMBs in the pattern generation process that is to be described next. In the proposed technique, if the number of '1's is less than 8 (same as in [13]), the MB has very low movement so that it can be encoded as skipped block. On the other hand, if the total number of '1's is more than 64 + 2QP/3, the MB has very high motion so that it can be encoded using standard H.264 modes. Obviously, more MBs are encoded using the pattern mode at low bit rates as compared to high bit rates. Thus, we also relate the upper-bound threshold with QP to regulate the number of CRMBs with different bit rates.

Once all such CRMBs are collected for a certain number of consecutive frames, decided by the rate-distortion optimizer [18] when the rate-distortion gain outweighs the overhead of encoding the shape of new patterns, these are divided into $\alpha$ sets to generate patterns. In order to generate patterns with minimal overlapping, a simpler greedy heuristic is employed where these CRMBs are divided into $\alpha$ clusters such that the average distance among the gravitational centres of CRMBs within a cluster is small while the same among the centres of CRMBs taken from different clusters is large. The CPG algorithm generates $\mu$-pixel pattern for a cluster by the $\mu$-most-frequent pixels among all the CRMBs in the cluster.

### 2.2. Encoding and decoding of PVC using content-based pattern codebook
The Lagrangian multiplier [19,20] is used to trade off between the quality of the compressed video and the bit rate generated for different modes. In this method, the Lagrangian multiplier, $\lambda$ is calculated with an empirical formula using the selected QP for every MB in the H.264 [18] as follows:

$$\lambda = 0.85 \times 2^{(QP-12)/3}. \tag{2}$$

During the encoding process, all possible modes including the pattern mode are first motion estimated and compensated for each MB, and the resultant rates and the distortions are determined. The final mode $m$ is selected as follows:

$$m_n = \arg\min(D(m_i) + \lambda B(m_i)) \tag{3}$$

Where $B(m_i)$ is the total bits for mode $m_i$, including mode type, motion vectors, extra pattern index code for pattern mode, and residual error after quantization, while $D(m_i)$ is measured as the *sum of square difference*

between the original MB and the corresponding reconstructed MB for mode $m_i$.

## 3. Proposed algorithm
As mentioned earlier, the CPG algorithm can generate an optimal pattern from given moving regions but there is no guarantee to generate optimal multiple patterns from the entire given set of moving regions. For simplicity, it uses a clustering technique which divides the moving regions into $\alpha$ clusters to generate $\alpha$ patterns. Thus, it is obvious that the performance of CPG also depends on the efficiency of clustering technique. As aforementioned, a clustering problem is a NP-complete problem and thus, a global optimization algorithm would be computationally unworkable. We propose a heuristic which can solve this problem near-optimally.

### 3.1. Optimal content-based pattern generation algorithm
Without losing any generality, we can assume that an optimal clustering technique with the CPG algorithm can provide optimal pattern codebook. We can define an optimal codebook, if each moving region is best-matched by the pattern which is generated from the cluster of that moving region. Suppose that an optimal clustering technique divides the CRMBs into clusters $C_1, C_2,...,C_\alpha$. If the pattern $P_i$ is generated from the $C_i$, i.e.,

$$P_i = CPG(C_i) \tag{4}$$

and the pattern $P_j$ is selected as the best matched pattern for the moving region $M \in C_i$ as

$$P_j = \arg\min_{P_n \in PC}(|M|_1 - |M \wedge P_n|_1) \tag{5}$$

then $P_i$ and $P_j$ will be same for an optimal pattern codebook, and $\wedge$ represents the AND operation.

In the actual coding phase, a CRMB of a cluster can be approximated by the following two approaches: the pattern generated from its corresponding cluster or the best matched-pattern from the pattern codebook irrespective of its clusters. The first approach is termed as *direct pattern selection* and later approach is *exhaustive pattern selection*.

The correct classification rate, $\tau$, can be defined as the fraction of the number of CRMBs matched by the pattern using direct pattern selection against entire CRMBs. Due to the overlapped regions of the patterns, there is a probability to better approximate a CRMB with a pattern generated other than its cluster. Obviously the probability of $\tau$ will increase with the number of patterns in a codebook due to the better similarity between moving region and the corresponding pattern. Moreover, a small number of patterns cannot better approximate the CRMBs, as a result there is

always a possibility of ignoring a CRMB using the pattern mode, if only the extracted pattern from a cluster is used to match against the CRMBs of the same cluster. Thus, this system requires reasonable number of patterns. On the other hand, we can called a CPG algorithm as the globally optimal one if it produces a pattern set in such a way that each CRMB is best-similarity-matched by the pattern which is generated from its own cluster, i.e., the value of $\tau$ is 100%. We can define $\tau$ as follows where $|CRMB_s|$ indicates the total number of CRMBs:

$$\tau = \left( \sum_{k=1}^{|CRMBs|} x(k) \begin{cases} x(k) = 1 \text{ if } P_i = P_j \\ 0 \text{ otherwise} \end{cases} \right) / |CRMBs| \quad (6)$$

where $P_i$ and $P_j$ are selected from Equations 4 and 5, respectively. When $\tau = 100\%$, we get the optimal solution using clustering and the CPG algorithm. To do this we need to modify the CPG algorithm where a generic clustering technique using pattern similarity metric as a part of this algorithm. The dissimilarity of a CRMB against a pattern, $P_n$ is defined as:

$$\psi^n(M) = |M|_1 - |M \wedge P_n|_1. \quad (7)$$

where $M$ and $P_n$ are the CRMB and the $n$th pattern respectively. The best-matched pattern is selected using Equation 5.

Unlike the CPG, optimal CPG (OCPG) (detailed in Figure 3) performs clustering and pattern formation until $\tau$ is 100% in each iteration. For a *seed* pattern codebook, it ensures that each CRMB will be best-matched by a pattern generated from its own cluster, i. e., the clustering process is optimum. However, it does not guarantee the global optimality of clustering because of trapping in local optima. To ensure the global optimality we need to determine average dissimilarity $\psi_{avg}$ using pattern codebooks generated with iterations. The final pattern codebook is selected based on the minimum $\psi_{avg}$ for a given number of iterations with random starts as $\psi_{avg}$ indicates the optimal pattern codebook. We can define $\psi_{avg}$ as follows where $C$ indicates the total set of CRMBs, $C_i$ indicates the $i$th sub-set of CRMBs clustered using $i$th pattern, $|C_i|$ indicates the total number of CRMBs in $C_i$, and $\psi^i(C_i(j))$ indicates the dissimilarity between $i$th pattern and $j$th CRMB in the $C_i$ sub-set:

$$\psi_{avg} = \left( \sum_{i=1}^{\alpha} \sum_{j=1}^{|C_i|} \psi^i(C_i(j)) \right) / \sum_{i}^{\alpha} |C_i|. \quad (8)$$

For one random start, we will get a candidate global solution for a seed codebook. There would be multiple solutions for given moving regions. When the search space is really large and there is no suitable algorithm to find the optimum solution, $k$-change neighbourhood may be considered as a $k$-optimal solution [21]. Lin and Kernighan [22] empirically found that a 3-optimal solution for the *travelling salesman problem* has a probability of about 0.05 of being not optimal, and hence 100 random starts yields the optimum with a probability of 0.99. Lin and Kernighan also demonstrated that a 3-optimal solution is much better than a 2-optimal solution; however, a 4-optimal solution is not sufficiently superior to a 3-optimal solution to justify the additional computational cost. In our approach we also use 100 random starts and replace 3-pixel in each pattern to get the optimal solution. We terminate each iteration of a random start when either the average dissimilarity is not reduced in successive iteration or $\tau = 100\%$. Thus, OCPG ensures convergence by providing near-optimal solutions.

The main advantage of this global OCPG approach over the local CPG approach is that it takes whole moving region information to cluster the CRMB against the pattern (instead of a gravitational centre of a CRMB [15]). Moreover, multiple iterations ensure the quality of pattern codebook to represent the CRMBs and this approach does not require exhaustive pattern matching so that it reduces the computational time needed to select the best-match pattern from a codebook against each CRMB.

Figure 4 shows the way to generate a pattern using the proposed OCPG algorithm. Figure 4a shows 3D representation of the total moving regions for the corresponding pixel position which is calculated by the summation of all CRMBs' '1's in a cluster in the first iteration. This 3D representation indicates the most significant moving area (where the frequency is high) in a cluster. Figure 4d shows the same thing after the final iteration. Note that Figure 4d has more concentrated high frequency area compared to Figure 4a, and this suggests the necessity of global optimization for pattern generation. Figure 4b, e show the 2D cluster view. The final patterns are shown in Figure 4c, f where the latter is obviously the desirable pattern due to the compactness.

### 3.2. Impact of OCPG algorithm on correct classification rate $\tau$, dissimilarity $\psi$, and number of iterations

Figure 5 shows average number of iterations needed for each random start to provide $\tau = 100\%$ using ten standard QCIF video sequences. The average is 9.73 per random start, would be much lower if we use seed patterns for each start. But the seed pattern may bias towards the seed pattern shape.

Figure 6 shows the 32 patterns used in [14,15]. To generate the arbitrary number of patterns using

**Algorithm** $PC = OCPG(\alpha, \mu, K, C)$

*Precondition: Given a set of CRMBs, C, Given iterations K;*

*Post condition: A pattern codebook* $PC = \{P_1,...,P_\alpha\}$ *of* $\mu$*-pixel*

*content-based patterns;*

1. $k = 0; \ \tau = 0; \ \psi_{avg} = \infty; \ Replace = 0;$

2. WHILE ($k < K$)

3.      Randomly generate $\alpha$ number of patterns, $P_1,...,P_\alpha$ of $\mu$-pixel

4.      Divide $C$ into $\alpha$ clusters based on the equation (5) using PC or any clustering algorithm.

5.      $t = 0; \ \tau = 0;$ Calculate $\psi_{avg}^P$ using current PC for all *MR*.

6.      WHILE ($\tau$<100)

7.          For $i=1,...,\alpha$

8.            For $x = 0,...,15$

9.              For $y = 0,...,15$

10.                $P_i(x,y) = 0;$

11.              $T_i(16x+y) = f_i(x,y) = \sum_{j=1}^{|C_i|} M_{i,j}(x,y)$ where

$M_{i,j}$ is the MR of the $j^{th}$ CRMB in $C_i$;

12.              $\{l_0,...,l_{255}\}$ =ranked indices of $T_i$ such that

$T_i(l_j) \geq T_i(l_{j+1})$ for $0 \leq j < 255;$

13.            For j=0,...,$\mu$-1

14.              $P_i(\lfloor l_j/16 \rfloor, l_j \bmod 16) = 1;$

15.          Divide $C$ into $\alpha$ clusters based on the equation (5) using new PC and calculate $\tau$ and $\psi_{avg}^C$ for all $M$.

16.          IF ($\psi_{avg}^C > \psi_{avg}^P$) exit; ELSE $\psi_{avg}^P = \psi_{avg}^C$;

17.          $t = t + 1;$

18.     IF $\psi_{avg} > \psi_{avg}^C$

19.          $\psi_{avg} = \psi_{avg}^C; \ PC = \{P_1,...,P_\alpha\}$

20.   $k = k + 1;$

**Figure 3 The OCPG algorithm for near optimal multiple pattern sequence generation**.



**Figure 4 Pattern generation using the proposed OCPG algorithm**. **(a, d)** 3D representation of pixel frequency of one of the eight clusters of CRMBs obtained from *Foreman* video sequence, for the first and last iterations, respectively. **(b, e)** Their corresponding 2D top view projection; and **(c, f)** the generated pattern for this cluster by the OCPG algorithm after first iteration and final iteration for a random initial seed pattern. Please refer to the text for more explanation.

definition, certain features are assumed for each 64-pixel pattern such that each is *regular* (i.e., bounded by straight lines), *clustered* (i.e., the pixels are connected), and *boundary-adjoined*. Since the moving region of a MB is normally a part of a rigid object, the clustered and boundary-adjoined features of a pattern can be easily justified, while the regularity feature is added to limit the pattern codebook size.

Figure 7 shows some example patterns from the seven test sequences. It is interesting to note the lack of similarity between the pattern sets for each of the sequences. The patterns cover different regions of a MB to ensure the maximum pattern coded MBs form maximum compression. It should also be noted that of the three fundamental pixel-based assumptions, which apply to any predefined codebook, only regularity has been relaxed, while the clustering and boundary-adjoined conditions are adhered to it in most cases. This relaxation is one of the main reasons for the
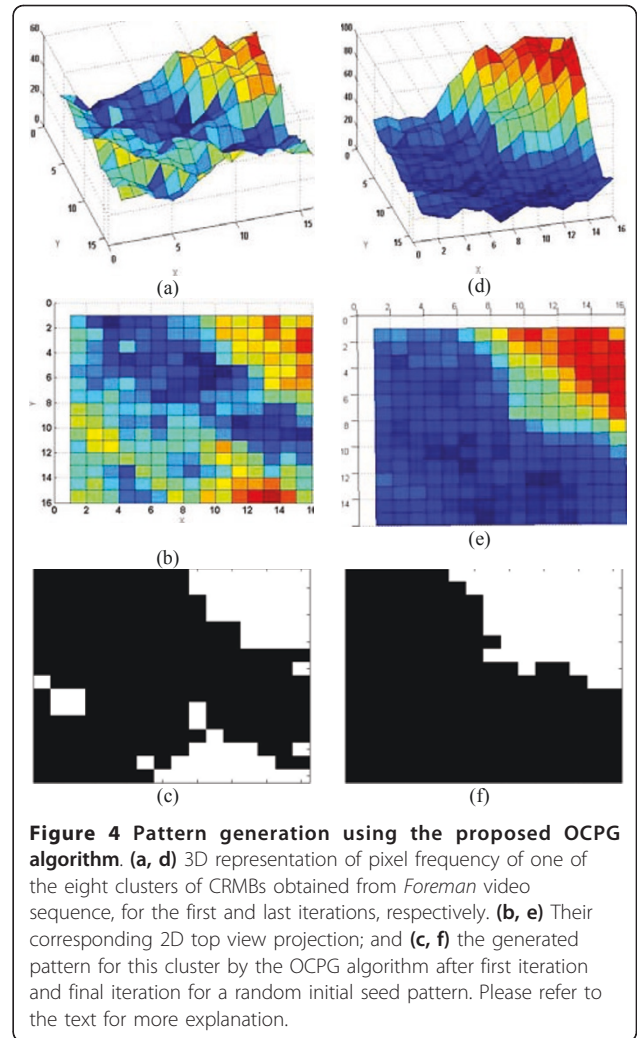
superior coding efficiency achieved by the arbitrary-shaped patterns.

Figure 8 shows that how the proposed OCPG algorithm generates the optimal codebook. For each random start the OCPG algorithm reduces the dissimilarity, $\psi_{avg}$
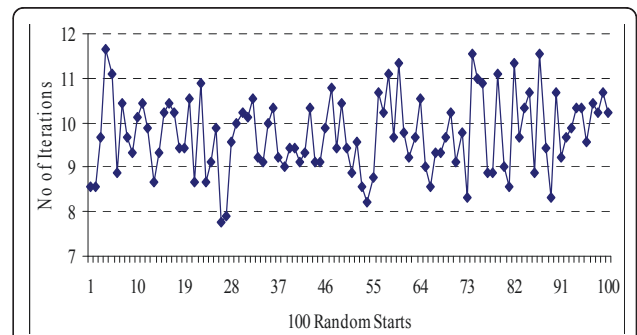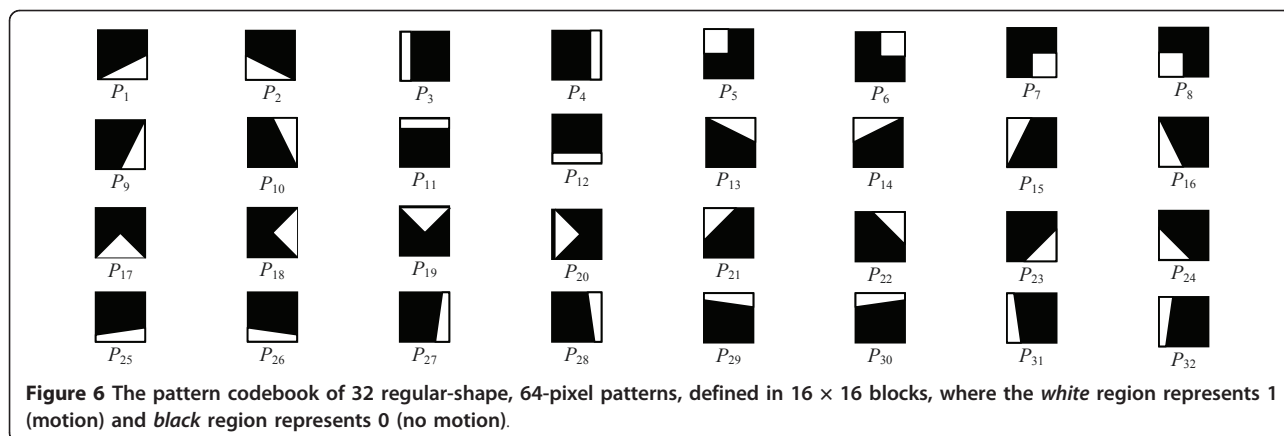


**Figure 5 Average number of iterations is needed to get** $\tau = 100\%$ **with 100 random starts using 10 standard QCIF video sequences where the total average is 9.73**.

**Figure 6 The pattern codebook of 32 regular-shape, 64-pixel patterns, defined in 16 × 16 blocks, where the *white* region represents 1 (motion) and *black* region represents 0 (no motion).**

(see Figure 8b) by classifying the CRMBs (Line 15 of the proposed OCPG algorithm) using the best-matched pattern. Thus, it increases $\tau$ (see Figure 8a) and also ensures the convergence of the OCPG algorithm.
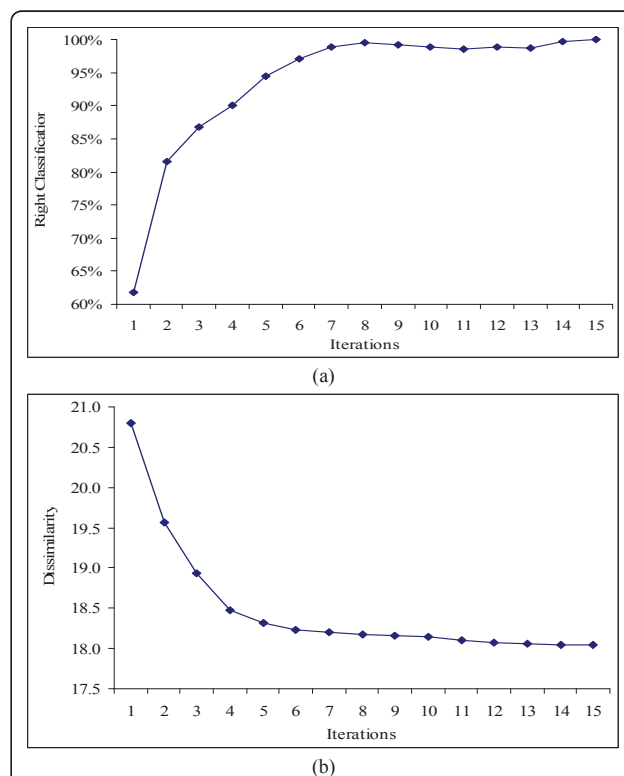
It is clear that the coding performance will decrease with the group of frames participated in the pattern formation process of the proposed OCPG algorithm as the generated PC is gradually approximating the shape of the CRMBs. This process imposes restriction on the group of frames size. Thus, we need to refresh the pattern codebook with a regular interval. As shown by experiments, the *group of picture* (GOP) size would be good candidate to test whether we need to refresh the codebook. The detailed procedure of pattern codebook refreshment and transmission will be described in Section 3.4.

### 3.3. Clustering techniques

The CPG algorithm uses *K*-means clustering technique [23] where it uses gravitational centre of the CRMBs to cluster them. The average value of $\tau$ is at best 70% using
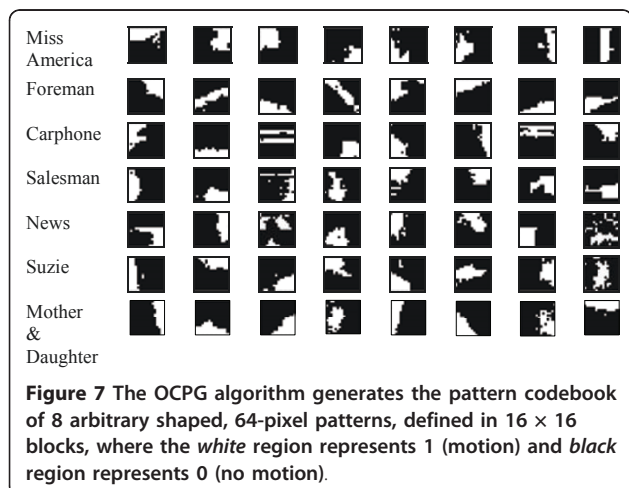
the CPG algorithm. It is due to the gravitational centre which represents all the 256 pixels with a point. We also investigate *Fuzzy C-means* [24,25] clustering technique, but the results is almost the same. Neural network is not a good candidate due to the computational complexity. It is interesting to note that the performance of the proposed OCPG algorithm does not depend on any specific clustering algorithm because whatever a



(a)



(b)

**Figure 8 Improvement of clustering process using the proposed OCPG algorithm for the best random start using the first GOP of *Miss America* sequence, where (a) $\tau$ increases and (b) $\psi_{avg}$ decreases with the iterations.**



**Figure 7 The OCPG algorithm generates the pattern codebook of 8 arbitrary shaped, 64-pixel patterns, defined in 16 × 16 blocks, where the *white* region represents 1 (motion) and *black* region represents 0 (no motion).**

clustering algorithm used is merely to generate only seed codebook and subsequently, the process converges quickly with our pattern similarity matching algorithm.

### 3.4. Pattern codebook refreshing and coding

For content-based pattern generation, we need to transmit the pattern codebook after a certain interval. To determine whether we need to transmit the newly generated codebook or continue with the current one, we consider the bits and distortions generated with both the current and the previous pattern codebooks. The GOP [26] may be the best choice as after a GOP we need to send a fresh *Intra* picture in the bitstream. Note that this GOP may be different from the group of frames used for codebook generation. To trade off the bitstream size and quality we can use Lagrangian optimization function as it is used to control the rate-distortion performance. Here we consider average distortion and bits per MB in both cases. We select the current codebook if it provides less Lagrangian cost as compared to the previous one.

From the experimental results we observe that around 2 to 4 times we need to refresh arbitrary patterns while we use first 100 frames of the seven standard QCIF video sequences (same as those used in Figure 7) as illustrated in Figure 9. The figure also shows that the number of transmission increases with the bit rate because almost fixed amount of bits for pattern transmission has significant contribution in rate-distortion optimization at the low bit rate but has insignificant contribution at relatively high bit rates. Note that five times in refreshment mean that we need to refresh the pattern codebook in all GOP in our experiments.

For pattern codebook transmission we have divided each pattern (i.e., $16 \times 16$ pixel binary MB) into four $8 \times 8$ blocks and then applied zero-run length coding.
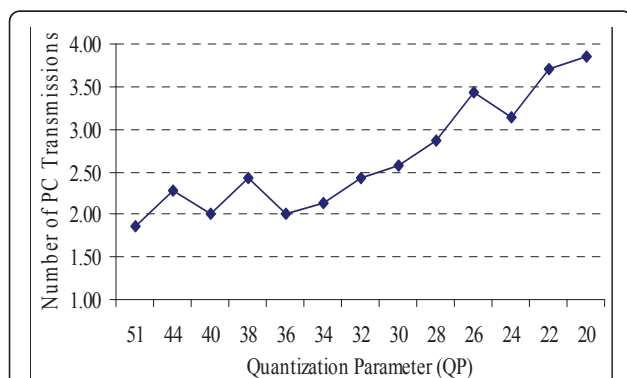


**Figure 9 The average number of pattern code transmissions with quantization parameters when we processed first 100 frames using seven standard QCIF video sequences**. namely *Miss America, Suzie, Claire, Salesman, Car phone, Foreman*, and *News* of 30 frames per second

The zero-run length will be 0-63 as the total number of elements in a block are 64. We have used *Huffman* coding to assign variable length codes for each combination. The length of codes varies from 2 to 14 bits for the length of 0-63. But we have treated the length of 64 (i. e., all are zeros) as a special case, and thus we assign it with two bits as well. As for the variable length codes, one can easily generate them from the frequencies of the zero-run lengths using *Huffman* coding, so we do not include the whole table in this paper. From the experimental results, for eight patterns with each of 64 '1's requires 518 bits on average. On the other hand, if we use fixed length coding for the positions of '1's in a pattern we need 4,096 bits to transmit eight patterns with 64 '1's (i.e., $8 \times 64 \times 8 = 4096$ bits).

### 3.5. Multiple pattern modes and allowance of all MBs as CRMBs

As we mentioned in "Introduction", the best pattern selection process relying on the similarity measures does not guarantee that the best pattern would always result in coding efficiency because of residual errors after quantization and the choice of Lagrangian multiplier. To address this we use multiple pattern modes that select the pattern in the order of similarity. Since the similarity measure is a good estimator, we only consider higher ranked patterns. Eliminating the CRMB classification threshold for $8 \leq |M|_1 < 2QP/3 + 64$, by allowing all MBs to be motion estimated and compensated using pattern modes and finally selected by the Lagrangian optimization function, provides better rate-distortion performance. Obviously this increases the computational complexity which is checked using already known motion vector determined by the $16 \times 16$ mode.

### 3.6. Encoding and decoding in the proposed technique

In the proposed technique, near-*global-optimal arbitrarily shaped PVC* (ASPVC-Global) uses a pattern codebook that comprises eight patterns with 64 pixels as '1'. Note that a pattern is a $16 \times 16$ binary MB with 64 positions are marked as '1' and the rest of the positions are marked as '0'. The proposed OCPG technique is generic to form any pixel-size patterns (for example 64 as used in the experiment, 128, or 192) with any number of patterns (for example 2, 4, 8 as used in the experiment, 16, 32) in a codebook. We have investigated into different combinations of pixels and patterns, but found that the eight 64-pixel patterns are the best pattern codebook in terms of rate-distortion and computational performance using different video sequences. We have used fixed length codes (i.e., 3 bits) to identify each pattern in the proposed technique. Note that we have also encoded the pattern mode using finer quantization. In the implementation we have used $QP_{pattern} = QP - 2$,

where $QP$ is used for the other standard modes. The rationality of the finer quantization is that as the pattern mode requires fewer bits as compared to the other modes, we can easily spend more bits in coding residual errors by lowering quantization. The final mode decision is taken place by the Lagrangian optimizer.

Before encoding a GOP a new pattern codebook is generated using all frames of the GOP. Then encode that GOP using the new codebook and the previous codebook (if there is one, for the first GOP there is no previous codebook). We have selected the bits stream based on the minimum cost function (using Equation 3 with new Lagrangian Multiplier (see Section 5.1) using the average bits and distortion (sum of square difference) per MB.

As mentioned earlier, we have used the motion vector of the 16 × 16 mode as the pattern mode motion vector to avoid computational time requirement of ME. Only the pattern-covered residual error (i.e., region marked as '1' in the pattern template) is encoded and the rest of the regions are copied from the motion translated region of the reference frame. To encode a pattern-covered region, we need four 4 × 4-pixel sub-blocks (as 64 '1's in a pattern) for DCT transformation. Using the existing shape of the pattern (for example, the first pattern in the *Miss America* video sequence in Figure 7), we may need more than four 4 × 4-pixel blocks for DCT transformation. To avoid this, we need to rearrange the 64 positions before transformation so that we do not need more than four blocks. Inverse arrangement is performed in the decoder with the corresponding pattern index, and thus we do not lose any information.

In the decoder, we can determine the pattern mode and the particular pattern from the MB type and pattern index codes respectively. From the transmitted pattern codebook, we also know the shape of the patterns i.e., the positions of '1's and '0's. After inversely arranging the residual errors according to the pattern, we reconstruct the MB of the current frame by adding residual error with the motion translated MB of the reference frame.

## 4. Computational complexity of OCPG algorithm
In order to determine the computational complexity of the proposed ASPVC-Global algorithm, let us compare it with the H.264 standard. From now on, previous content-based PVC is named as ASPVC-Local [16]. The H.264 encodes each MB with motion search for each mode. When the proposed ASPVC-Global scheme is embedded into the H.264 as an extra mode, additional one-fourth motion search is required per MB as the pattern size is a quarter of a macroblock. Each macroblock takes part in the proposed OCPG algorithm and the best pattern is selected at the end. For detailed analysis

of the proposed OCPG algorithm is described as follows.

We can divide the entire process into (i) Binary matrix calculation, (ii) clustering and correct classification rate $\tau$ calculation, (iii) pixel frequency calculation of each cluster, and (iv) sorting the pixels based on the frequency. Let $N$, $\alpha$, $M^2$, $k$, and $I$ be the total number of MBs, total number of cluster, block size, total number of random starts, and number of iterations for $\tau = 100\%$, respectively, and then:

[i] Each binary matrix calculation requires one subtraction, one absolute and one comparison. Thus, totally $3NM^2$ operations are required.

[ii] Each clustering requires one comparison and one addition. Thus totally $2\alpha NM^2$ operations are required. Each correct classification rate calculation requires one comparison. Thus, $\alpha N$ operations are required.

[iii] Each pixel frequency calculation requires one addition. Thus $NM^2$ operations are required.

[iv] Sorting the pixel frequency requires $2\alpha M^2 \ln M$ operations.

Therefore, the proposed OCPG algorithm requires $3NM^2 + kI(2\alpha NM^2 + \alpha N + NM^2 + 2\alpha M^2 \ln M)$ operations. If we assume that $N >> \alpha$ and $N >> M$, the required operations would be $nM^2(4 + 16K)$ where $K$ is the total number of iterations including the number of random starts and the associated inner-loop iterations. On the other hand, motion search using any mode requires $3(2d + 1)^2 NM^2$ operations where $d$ is the range of motion search. Thus, the proposed ASPVC-Global with 100 random starts and 9.73 (according to Figure 5) inner-loop iterations until $\tau = 100\%$ requires no more than 5.4 times operations compared to the full motion search by a mode where search length is 15.

Compared to the fractional as well as multi-mode motion search this extra operation does not restrict it from real time operations. The experimental results also show that maximum of dissimilarity is within 7% of the minimum dissimilarity of 100 random starts. It means that if we consider only one start, we only lose 7% of clustering accuracy. Thus, according to the availability of computing power or hardware, we can make the proposed OCPG efficient by reducing the number of random starts. The experimental results show that with only five random starts we can achieve very similar performance of optimal one and much better than the existing approach. The OCPG with five random starts and 9.73 iterations for $\tau = 100\%$ requires no more than 30% of more operations compared to the full motion search using a mode where search range is 15 pixels.

For multiple pattern modes, the ASPVC-Global needs only bit and distortion calculation without ME. The ME, irrespective of a scene's complexity, typically comprises more than 60% of the computational overhead required to encode an inter picture with a software codec using the DCT [27,28], when full search is used. Thus, maximum of 10% operations are needed for one pattern mode as each pattern mode will process one-fourth of the MB. As a result, the ASPVC-Global algorithm using five random starts and up to four pattern modes may requires extra 0.58 of a mode ME&MC operations compared to the H.264 which would not be a problem in real time processing.

## 5. Experimental set up and simulation results

### 5.1. Integration with H.264 coder

To accommodate extra pattern modes in the H.264 video coding standard for testing, we need to modify its bitstream structure and Lagrangian multiplier. For inclusion of pattern mode we change the header information for MB type, pattern identification code, and shape of patterns. Inclusion of pattern mode also demands modification of the Lagrangian multiplier as the pattern mode is biased to bits rather than distortion.

The H.264 recommendation document [3] provided binarization for MB and sub-MB in $P$ and $SP$ slices. Experimental results show that in most of the cases the $8 \times 8$ mode is less frequent compared to the larger modes. Thus, we use first part of the MB type header for the pattern mode using '001' code and then assign variable length codes for pattern modes, $8 \times 8$, $8 \times 4$, $4 \times 8$, and $4 \times 4$. Using the frequency of MB type, we assigned the pattern modes, $8 \times 8$, $8 \times 4$, $4 \times 8$, and $4 \times 4$ as '0', '10', '111', '1100', and '1101', respectively. After the header of MB type we need to send the pattern type with the maximum length of codes as $\log_2$(number of pattern templates) when fixed length pattern codes will be used. For example, when we use eight patterns in a codebook, we use 3 bits for the pattern code. The pattern code will identify the particular pattern. At the beginning of a GOP we transmit the codebook if necessary. We use one bit to indicate whether a new codebook is transmitting.

We also investigate into Lagrangian multiplier after embedding new pattern modes in the H.264 coder. It is already mentioned earlier that a new pattern mode yields less bits and sometimes higher distortion compared to the standard H.264 modes. To be fair with the other modes, the value of multiplier is reduced to $\lambda = 0.4 \times 2^{(QP-12)/3}$. The experimental results of Lagrangian multiplier and rate-distortion performance have justified the new valuation. As the pattern modes require fewer bits compared to the $16 \times 16$ mode, the reduced $\lambda$ signifies less importance in bits as compared to the

distortion in the minimization of Lagrangian cost function. We have also observed that for a given $\lambda$, the generated $QP$ is slightly large for relatively high motion compared to the smooth motion video sequences.
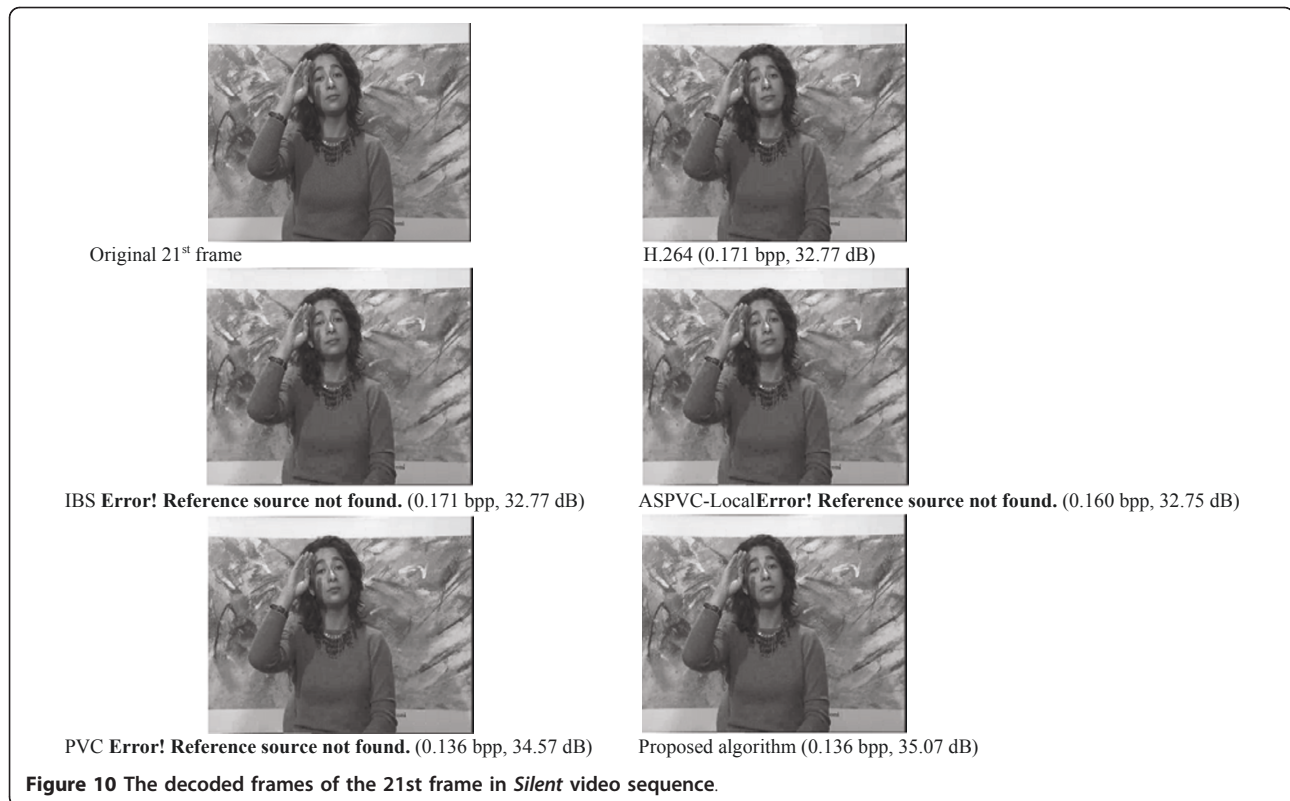
### 5.2. Experiments and results

In this paper, experimental results are presented using nine standard video sequences with wide range of motions (i.e., smooth to high motions) and resolutions (QCIF to 4CIF) [26]. Among them, three (*Miss America*, *Foreman*, and *Table Tennis*) are QCIF (176 × 144), one (*Football*) is SIF (352 × 240), two (*Paris* and *Silent*) are CIF (352 × 288), and other two (*Susie* and *Popple*) are 4CIF (720 × 576). Full-search ME with 15 as the search range and fractional accuracy has been employed. We have selected a number of existing techniques to compare with the proposed one. They are the H.264 (as it is the state-of-the-art video coding standard), the ASPVC-Local [16] (as it is the latest block partitioning coding technique with arbitrarily shaped patterns), the IBS [12] (as it is the latest block partitioning video coding technique), and the PVC [15] (as it is the latest block partitioning technique using pre-defined patterns).

Figure 10 shows some decoded frames for visual viewing comparison by the H.264, the IBS [12], the ASPVC-Local [16], PVC [15], and the proposed techniques. The 21st frame of *Silent* sequence is shown as an example. They are encoded using 0.171, 0.171, 0.160, 0.136, and 0.136 bits per pixel (bpp) and resulting in 32.77, 32.77, 32.75, 34.57, and 35.07 dB in Y-PSNR, respectively. Better visual quality can be observed in the decoded frame constructed by the proposed technique at the fingers areas. Apart from the best PSNR result by the proposed technique, subjective viewing has also confirmed the quality improvement. From the viewing tests with 10 people, the decoded video by the proposed scheme is with the best subjective quality. It is due to the fact that the proposed method performs well in the pattern-covered moving areas, and the bit saving for partially skipped blocks (i.e., exploiting more of intra-block temporal redundancy) compared to the other methods. Thus, the quality of the moving areas (i.e., area comprising objects) is better in the proposed method.

Table 1 shows rate-distortion performance for a fixed bit rate using different algorithms for different video sequences. The table reveals that the proposed algorithm outperforms the relevant existing algorithms such as H.264, the IBS [12], the ASPVC-Local [16], and the PVC [15] by 2.2, 2.0, 1.5, and 0.5 dB, respectively.

Figure 11 shows overall rate-distortion performance for wide range of bit rates using different types of video sequences (in terms of motion and resolution) by the H.264, the IBS [12], the ASPVC-Local [16], the PVC [15], and the proposed techniques. For all cases, the

Original 21$^{st}$ frame

H.264 (0.171 bpp, 32.77 dB)

IBS **Error! Reference source not found.** (0.171 bpp, 32.77 dB)

ASPVC-Local**Error! Reference source not found.** (0.160 bpp, 32.75 dB)

PVC **Error! Reference source not found.** (0.136 bpp, 34.57 dB)

Proposed algorithm (0.136 bpp, 35.07 dB)

**Figure 10 The decoded frames of the 21st frame in *Silent* video sequence**.

proposed technique outperforms the state-of-arts techniques. The proposed technique outperforms the most recent PVC technique [15] by at least 0.5 dB for almost all video sequences with wide range of bit rates. The proposed technique exhibits better performance due to the global optimization, allowing all MBs into multiple pattern generation and pattern modes, and spending more bits in pattern mode.

**Table 1 Performance at a glance**

| Video sequence @ kbps | H.264 | IBS | ASPVC-Local | PVC | Proposed |
|---|---|---|---|---|---|
| Miss America QCIF @72 | 37.0 | 37.2 | 38.6 | 39.7 | 40.3 |
| Table QCIF @200 | 32.2 | 32.2 | 32.2 | 32.7 | 33.0 |
| Foreman QCIF @200 | 32.8 | 32.9 | 33.1 | 33.6 | 34.1 |
| Mother&Daughter QCIF @110 | 34.4 | 34.4 | 35.2 | 36.8 | 37.2 |
| News QCIF @110 | 29.0 | 29.0 | 30.4 | 33.0 | 33.6 |
| Hall CIF @500 | 33.4 | 33.4 | 34.6 | 36.1 | 36.6 |
| Football SIF @1100 | 28.6 | 28.6 | 28.6 | 28.9 | 29.1 |
| Paris CIF @1100 | 34.5 | 34.5 | 34.7 | 35.8 | 36.5 |
| Silent CIF @600 | 33.6 | 33.6 | 33.8 | 35.6 | 36.3 |
| Table 4CIF @3500 | 29.6 | 29.7 | 29.8 | 30.2 | 30.7 |
| Tempate 4CIF @3500 | 32.4 | 32.4 | 32.4 | 32.7 | 33.1 |
| Popple 4CIF @3500 | 30.5 | 30.6 | 31 | 31.9 | 32.4 |

The performance of the proposed technique as well as other pattern-based video coding may not perform better significantly compared to the H.264 at high bit rates as the number of MBs encoded by the pattern-mode may diminish. It is due to the dominancy of the smaller modes of the variable block size over pattern mode. It may also fail if the video sequences have extremely high motion. It is due to the smaller amount of intra-block temporal redundancy available in MBs in such situations. After all, the proposed technique is good at low bit rates by the nature of its theoretical ground. It has been demonstrated above that its objectives have been achieved.

## 6. Conclusions

In this paper, we have proposed an efficient video coding technique using arbitrarily shaped block partitions in global optimal perspective, for low bit rates. The proposed scheme uses a content-based pattern generation strategy in the globally optimal perspective, based upon multiple pattern modes. A Lagrangian multiplier has been derived to embed the pattern mode into the H.264. We have verified the effectiveness of the proposed technique by comparing other contemporary and relevant algorithms. The experimental results show that this new scheme improves the video quality by 0.5 and
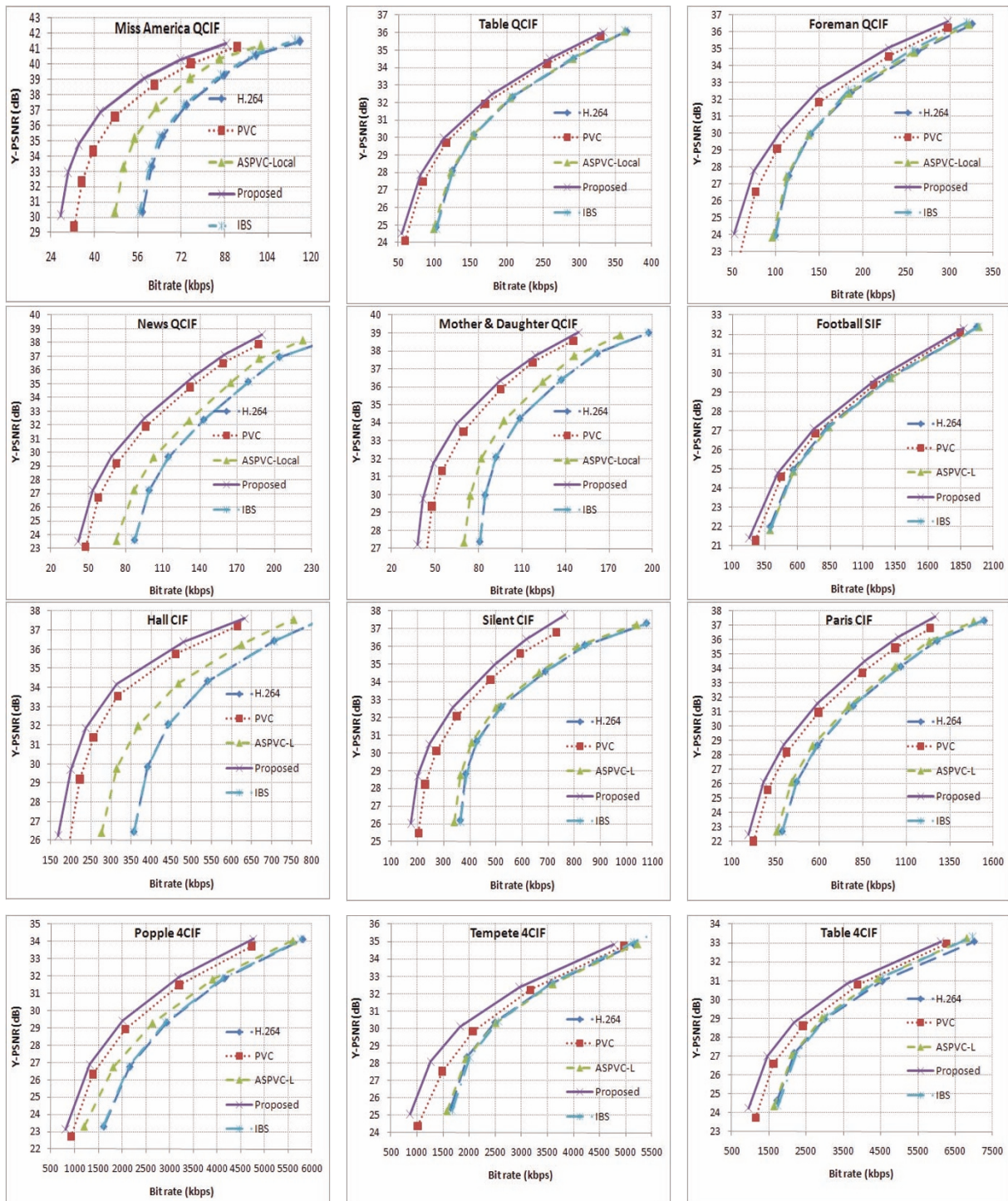
**Figure 11 Rate-distortion performance on standard video sequences using the proposed**. IBS [12], ASPVC-Local (i.e., ASPVC-L) [16], PVC [15], and the H.264 techniques.

1.5 dB compared to the existing latest pattern-based video coding and the H.264 standard respectively.

## Author's information

Additional email addres for Professor Paul: mpaul@csu.edu.au

## Abbreviations

ASPVC: arbitrarily shaped pattern-based video coding; BPP: bits per pixel; CPG: Content-based pattern generation; CRMB: candidate region-active macroblock; GOP: Group of picture; IBS: implicit block segmentation; ITR: intra-block temporal redundancy; MB: Macroblock; MC: motion compensation; ME: motion estimation; NP: non-polynomial; OCPG: optimal content-based pattern generation; PVC: pattern-based video coding; QP: quantization parameter.

## Author details

[1]School of Computing and Mathematics, Charles Sturt University, Panorama Avenue, Bathurst, NSW 2795, Australia [2]Gippsland School of Information Technology, Monash University, Churchill, VIC 3842, Australia

## Competing interests

A significant portion of the research work is done when I was a PhD student and research fellow in Monash University under the supervision of Manzur Murshed. I wrote this paper when I was not in Monash University. I have submitted this paper and modified the paper when I am a Lecturer in Charles Sturt University. Article processing fee is provided by Charles Sturt University.

## References

1. ITU-T Recommendation H.263. Video coding for low bit-rate communication, version 2 (1998)
2. ISO/IEC 13818, MPEG-2 International Standard (1995)
3. ITU-T Rec. H.264/ISO/IEC 14496-10 AVC. Joint Video Team (JVT) of ISO MPEG and ITU-T VCEG, JVT-G050 (2003)
4. M Paul, MM Murshed, Superior VLBR video coding using pattern templates for moving objects instead of variable-bloc size in H.264, in *the 7th IEEE Int Conferen Signal Proce (ICSP-04)*, (Beijing, China, 2004), pp. 717–720
5. P Li, W Lin, XK Yang, Analysis of H.264/AVC and an associated rate control scheme. J Electron Imaging. **17**(4), 043023 (2008). doi:10.1117/1.3036181
6. S Chen, Q Sun, X Wu, L Yu, L-shaped segmentations in motion-compensated prediction of H.264, in *IEEE Conference on Circuits and Systems (ISCAS-08)* (2008)
7. EM Hung, L Ricardo, D Queiroz, D Mukherjee, On MB partition for motion compensation, in *IEEE International Conference on Imaging Process (ICIP-06)*, pp. 1697–1700 (2006)
8. O Divorra-Escoda, P Yin, C Dai, X Li, Geometry-adaptive block partitioning for video coding, in *IEEE International Conference on Acoustic Speech, and Signal Processing (ICASSP-07)*, pp. I-657–I-660 (2007)
9. O Divorra-Escoda, P Yin, C Gomila, Hierarchical B-frame results on geometry-adaptive block partitioning, in *VCEG-AH16 Proposal, ITU/SG16/Q6/VCEG*, (Antalya, Turkey, January 2008)
10. T Fukuhara, K Asai, T Murakami, Very low bit-rate video coding with block partitioning and adaptive selection of two time-differential frame memories. IEEE Trans Circ Syst Video Technol , 7: 212–220 (1997)
11. J Chen, S Lee, K-H Lee, W-J Han, Object boundary based motion partition for video coding, in *Picture Coding Symposium* (2007)
12. JH Kim, A Ortega, P Yin, P Pandit, C Gomila, Motion compensation based on implicit block segmentation, in *IEEE International Conference on Image Processing (ICIP-08)* (2008)
13. K-W Wong, K-M Lam, W-C Siu, An efficient low bit-rate video-coding algorithm focusing on moving regions. IEEE Trans Circ Syst Video Technol. **11**(10), 1128–1134 (2001). doi:10.1109/76.954499
14. M Paul, M Murshed, L Dooley, A real-time pattern selection algorithm for very low bit-rate video coding using relevance and similarity metrics. IEEE Trans Circ Syst Video Technol. **15**(6), 753–761 (2005)
15. M Paul, M Murshed, Video coding focusing on block partitioning and occlusions. IEEE Trans Image Process. **19**(3), 691–701 (2010)
16. M Paul, M Murshed, An optimal content-based pattern generation algorithm. IEEE Signal Process Lett. **14**(12), 904–907 (2007)
17. P Maragos, Tutorial on advances in morphological image processing and analysis. Opt Eng. **26**(7), 623–632 (1987)
18. T Wiegand, H Schwarz, A Joch, F Kossentini, Rate-constrained coder control and comparison of video coding standards. IEEE Trans Circ Syst Video Technol. **13**(7), 688–702 (2003). doi:10.1109/TCSVT.2003.815168
19. GI Sullivan, T Wiegand, Rate-distortion optimization for video compression. IEEE Signal Process Mag. **15**, 74–90 (1998). doi:10.1109/79.733497
20. T Wiegand, B Girod, Lagrange multiplier selection in hybrid video coder control, in *IEEE International Conference on Image Processing (IEEE ICIP-01)*, pp. 542–545 (2001)
21. CH Papadimitriou, K Steiglitz, in *Combinatorial Optimization: Algorithms and Complexity*, (Prentice-Hall, India, 1939)
22. S Lin, BW Kernighan, An effective heuristic procedure for the traveling-salesman problem. Oper Res. **21**, 498–516 (1973). doi:10.1287/opre.21.2.498
23. JB MacQueen, Some methods for classification and analysis of multivariate observations, in *Proceeding of 5th Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1. (University of California Press, 1967), pp. 281–297
24. JC Dunn, A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. J Cybern. **3**, 32–57 (1973). doi:10.1080/01969727308546046
25. JC Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, (Plenum Press, New York, 1981)
26. IEG Richardson, H 264 and MPEG-4 Video Compression, (WIL, 2003)
27. T Shanableh, M Ghanbari, Heterogeneous video transcoding to lower spatio-temporal resolutions and different encoding formats. IEEE Trans Multimedia. **2**(2), 101–110 (2000). doi:10.1109/6046.845014
28. M Paul, W Lin, CT Lau, B-S Lee, Direct inter-mode selection for H.264 video coding using phase correlation. IEEE Trans Image Processing. **20**(2), 461–473 (2011)