An Efficient Low Bit-Rate Video-Coding Algorithm Focusing on Moving Regions

Kwok-Wai Wong, Kin-Man Lam, and Wan-Chi Siu

***Abstract*—Block-based motion estimation and compensation are the most popular techniques for video coding. However, as the shape and the structure of an object in a picture are arbitrary, the performance of such conventional block-based methods may not be satisfactory. In this paper, a very low bit-rate video coding algorithm that focuses on moving regions is proposed. The aim is to improve the coding performance, which gives better subjective and objective quality than that of the conventional coding methods at the same bit rate. Eight patterns are pre-defined to approximate the moving regions in a macroblock. The patterns are then used for motion estimation and compensation to reduce the prediction errors. Furthermore, in order to increase the compression performance, the residual errors of a macroblock are rearranged into a block with no significant increase of high-order DCT coefficients. As a result, both the prediction efficiency and the compression efficiency are improved.**

***Index Terms*—Motion estimation and compensation, moving region detection, video coding.**

I. Introduction

RECENTLY, the demand for applications of the digital video communication, such as video conferencing, videophone, and high-definition television, has increased considerably. However, the transmission rates over public switched telephone networks (PSTN) are very limited. Therefore, very low bit-rate video coding is an important technology for such applications. ITU-T recommendation H.263 [1] and H.263+ [2] are the successful international standards for video compression using block-based techniques. The coding structure of H.263 is based on H.261 [3]. In these standards, motion estimation and compensation are used to reduce temporal redundancies, and discrete cosine transform (DCT) is then applied to encode the motion-compensated prediction difference. The quantized DCT coefficients, motion vectors, and the side information are entropy coded using variable-length codes (VLCs). The key differences between these two coding schemes are the target bit rates, the supported picture formats, the precision of motion estimation, the VLCs, and the motion-vector coding.

In fact, motion estimation is the most important process for video coding. A precise motion predictor can achieve a good performance with image quality, as well as a high compression ratio. However, motion estimation is also the most time-consuming process, so a fast and precise motion predictor is necessary. A number of fast motion-estimation algorithms have been reported [4]–[7], which can achieve better results than that of conventional fast search algorithms, such as the three-step search, the 2-D logarithmic search, the conjugate direction search, etc. In our approach, an efficient method for motion estimation that focuses on moving regions is devised.

As the existing block-based coding schemes, such as H.263, do not take into account the arbitrary shape of moving objects, their prediction efficiency will not be as good as expected. In order to reduce the prediction error for the arbitrary-shaped object, switching coders were reported [8]–[11]. In the switching coder, model-based and H.261/H.263 coders are employed for encoding the video. The working principle is to switch between the two coders to generate a compensated image for encoding. These switching coders might outperform the one using the H.261/H.263 coder alone, but the model-based coded image might be discarded due to the accumulation of modeling errors. Takahiro *et al.* [12] proposed a block-partitioning method for improving the prediction efficiency. Based on four pre-defined patterns, a macroblock is divided into two parts in which motion estimation is performed separately. This approach gives a better performance compared to the H.263, but the complexity of motion estimation and compensation might be too heavy for real-time applications. Also, the four patterns might not be sufficient to represent the moving objects for practical applications. In this paper, an efficient encoding method with less complexity, using eight pre-defined patterns, is proposed.

The basic encoding and decoding structures of our approach are based on H.263. It includes motion estimation and compensation, DCT transformation, and quantization. In our approach, the moving regions in a frame are detected and then partitioned into macroblocks. One of the eight pre-defined patterns will be used to represent the moving regions for the purpose of motion estimation and compensation. The residual errors of the macroblock will be rearranged into a block without a significant increase in high-order DCT coefficients. However, if the patterns are insufficient to represent the moving regions in a macroblock, the conventional DCT-based coding method will be employed. Experimental results show that both the picture quality and the run time are improved.

II. Low Bit-Rate Video Coding Using Patterns

A frame in a sequence may consist of moving regions and static regions. It is unnecessary to encode the static regions in a frame, as they can be obtained from the reference frames directly. The moving regions should be encoded precisely; this is important for visual quality. In our approach, the moving regions in a frame are detected, and one of the pre-defined patterns is used to encode the moving region in a macroblock. The details of our algorithm are described as follows.

Manuscript received August 15, 2000; revised June 22, 2001. This paper was recommended by Associate Editor K. Aizawa.

The authors are with the Centre for Multimedia Signal Processing, Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong.

Publisher Item Identifier S 1051-8215(01)09154-6.

TABLE I
THE RULE FOR CLASSIFYING THE TYPES OF MB

Macroblock Type	SMB	AMB	RMB
Condition(s) for Macroblock Type Decision	$\sum_{i,j=0}^{15} M_K(i,j) < T_{SMB}$	$\left(\sum_{i1,j1=0}^{7} M_{K1}(i1,j1) \neq 0, \text{ and } \sum_{i2,j2=0}^{7} M_{K2}(i2,j2) \neq 0, \text{ and } \sum_{i3,j3=0}^{7} M_{K3}(i3,j3) \neq 0, \text{ and } \sum_{i4,j4=0}^{7} M_{K4}(i4,j4) \neq 0 \right)$ or $\min(D_{K,N}) > T_{RMB}$	$\min(D_{K,N}) < T_{RMB}$

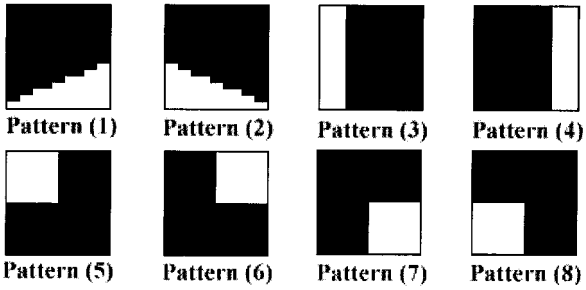

Fig. 1. The eight patterns for moving region approximation.

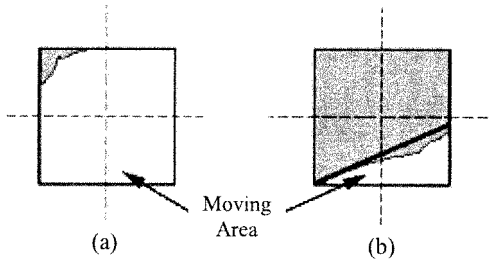

Fig. 2. Block type selection: (a) an AMB and (b) a RMB approximated by pattern (1).

A. Moving Region Detection

The pre-defined eight patterns which approximate the moving region in a macroblock are shown in Fig. 1. The white areas represent the moving region, while the black areas are the static region. The pre-defined patterns are obtained by investigating a large number of training patterns for encoding. The most popular eight patterns were selected and used for motion estimation and compensation in our approach. The moving region in a frame is detected by comparing the current frame $C(x, y)$ to its previous frame $P(x, y)$. The moving region $M(x, y)$ in a frame is obtained as follows:

$$M(x, y) = (T\,|C(x, y) \bullet B - P(x, y) \bullet B|) \qquad (1)$$

where $T(\cdot)$ is a thresholding function and B is the structuring element of morphological operations [13], [14]. The structuring element B is a square pattern of size 3×3. Performing the closing operation of a frame by the structuring element can reduce the noise in the frame. A pixel at position (x, y) will then be declared in a moving region if the difference of the processed current frame $C(x, y) \bullet B$ and the processed previous frame $P(x, y) \bullet B$ is greater than that of a threshold of value 2. Otherwise, the value of $M(x, y)$ is assumed to be zero.

From (1), the computed value of the static regions in a frame is zero, while the value of the moving regions is nonzero. The processed frame is then divided into macroblocks (MBs) for further classification. Three types of MB are defined in our approach: static MBs (SMBs), active MBs (AMBs), and active-region MBs (RMBs). If the contents of the MB are all zero or the size of moving region is smaller than a threshold, it is assumed to be a SMB. Otherwise, the MB will be divided into four sub-blocks for further classification. If the contents of all the sub-blocks have a nonzero value, the MB is defined as an AMB. The remaining MBs are assumed to be candidates of RMB. In this case, one of the eight pre-defined patterns will be used to approximate the moving regions. The best match pattern is obtained by finding a pattern that has the minimum value of $D_{K,N}$, as shown in the following function:

$$D_{K,N} = \frac{1}{256} \sum_{i=0}^{15} \sum_{j=0}^{15} |M_K(i, j) - P_N(i, j)| \qquad (2)$$

where M_K represents the Kth macroblock in the processed frame $M(x, y)$, and $P_N(i, j)$ represents the pre-defined pattern number N. The value of the patterns for moving regions is defined as one, while that of the static regions is zero. If the computed $D_{K,N}$ is greater than a threshold, this implies that the patterns are not good enough to represent the moving region. In this case, the MB is assumed to be an AMB, and the conventional motion estimation and DCT-based methods will be employed. Otherwise, the MB is declared to be a RMB, and one of the pre-defined patterns will be used for encoding. The rule for classifying the types of macroblocks is shown in Table I.

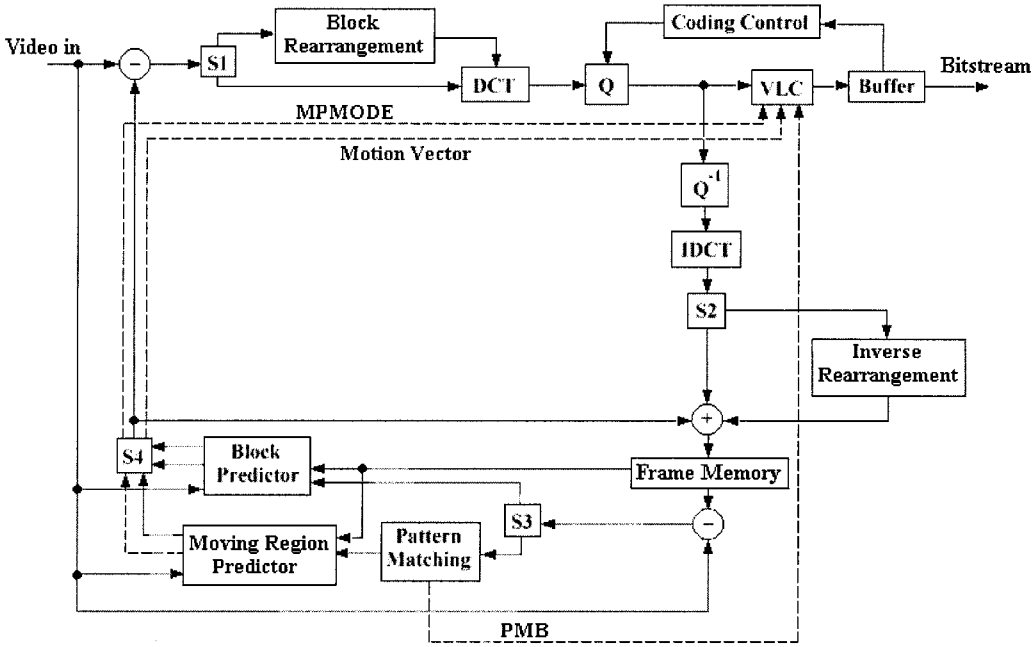

Fig. 3. Block diagram of the encoder.

The M_{K1}, M_{K2}, M_{K3}, and M_{K4} in Table I represent the four sub-blocks of M_K, T_{SMB} and T_{RMB} are the thresholds for macroblock type classification. In our approach, the threshold setting of T_{SMB} and T_{RMB} are 8 and 10, respectively. The function of T_{SMB} is to verify if there is no significant movement inside a MB, while T_{RMB} is used to determine a suitable pre-defined pattern for approximating the moving region. Examples of an AMB and a RMB are illustrated in Fig. 2(a) and (b), respectively.

B. Architecture of the Encoder and Decoder

The block diagram of the proposed video encoder is shown in Fig. 3. The structure is similar to a conventional video encoder, but includes additional features for encoding the moving regions:

1) *Type of MB:* A MB will be classified according to the results of moving region detection. Different encoding methods will be applied for different types of macroblock. If a SMB is detected, it is encoded as a skipped MB, which is supported by H.263. For the AMB, the conventional block-based encoding method is used. This includes block motion estimation, motion vector encoding, and residual error encoding. If RMB is detected, the best match pre-defined pattern will be extracted by the "Pattern Matching " unit for motion estimation and compensation.
2) *Pattern Matching:* The function of the "Pattern Matching" unit is to find a pre-defined pattern which is the best representation of the moving region in a RMB. The PMB (pre-defined pattern information) will be generated from this unit, and then encoded by VLC.
3) *Block Rearrangement:* If the input is a RMB, the switch S1 will be connected to the "Block Rearrangement" unit to rearrange the residual errors of the moving regions into a block of size 8×8. Hence, fast DCT algorithms can be used to encode the block. Similarly, the output from the IDCT will be rearranged in an inverse manner by connecting the switch S2 to the "Inverse Rearrangement" unit.
4) *Motion Estimation:* The respective prediction procedures for AMB and RMB are different. The switch S3 is connected to the upper path for AMB, while it is connected to the lower path for RMB, to which moving regions prediction is applied. The block predictor employs the conventional block-based motion estimation for encoding the AMBs. For RMB, the proposed moving region predictor as described in Section II-C is used for finding the motion vector which best represents the motion of the moving region. The static region is obtained directly from the reference frame. According to the MPMODE (motion predictor mode), the switch S4 will select the compensated image from either the block predictor or the moving region predictor. The MPMODE together with the motion vector are encoded by the VLC.

The architecture of the decoder is shown in Fig. 4. When the MPMODE indicates the current input MB to be a RMB, the switch S1 will connect to the "Moving Region Predictor" to generate the compensated image. The switch S2 will also select the upper path, the "Inverse Rearrangement" unit, to rearrange the residual error. For the AMB, the conventional block predictor will be employed, and the residual error will be decoded by inverse quantization, and then inverse DCT.

C. Motion Estimation and Compensation

The coding performance, which includes the picture quality, the run time, and the compression ratio, is affected by the motion estimation and compensation process. In our approach, the contents of SMB are copied from the reference frame directly. For

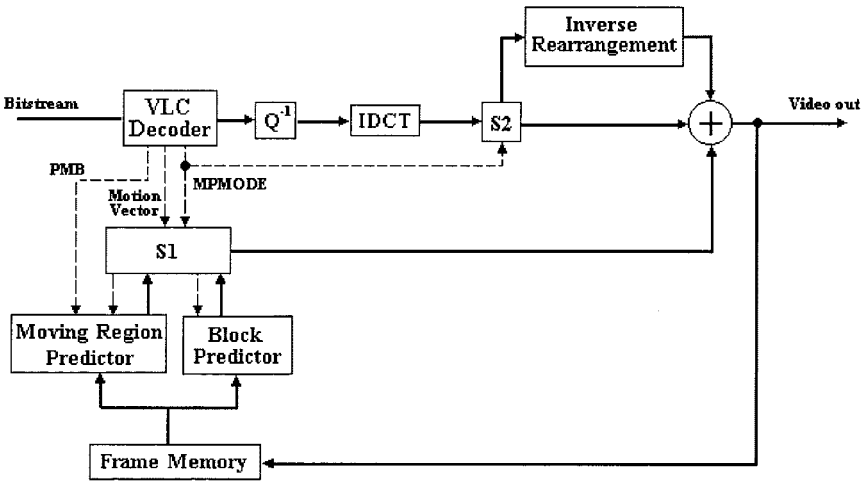

Fig. 4. Block diagram of the decoder.

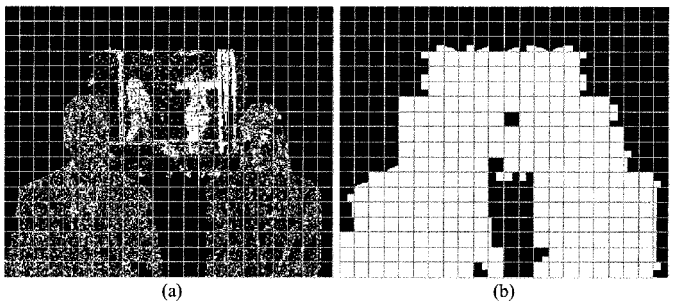

Fig. 5. An example of pattern approximation for the sequence "News": (a) detected moving regions and (b) results of pattern approximation.

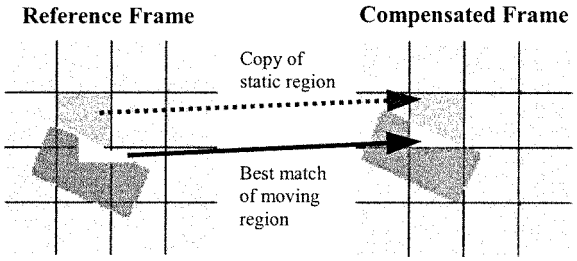

Fig. 6. The principle of interframe coding for RMB.

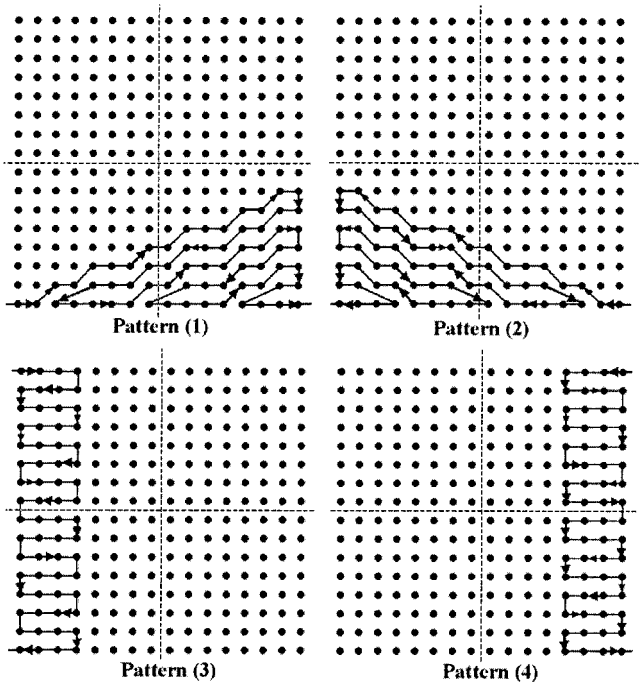

Fig. 7. Rearrangement of the residual errors.

the AMB, the conventional motion estimation and DCT-based methods are used. For RMB, both moving and static regions exist, but it is not necessary to use the static region in motion estimation. In [15], an efficient method was proposed for reducing the missing edge effect by using the edge information. In our approach, the motion vectors for the RMB are obtained by considering the moving regions only, which can reduce the complexity of the motion estimation process as well as prevent the missing edge effect. The matching process is more precise as the static region is neglected in motion estimation.

The mean-absolute difference (MAD) criterion function is employed for motion estimation. The motion vector for the moving region in a RMB can be obtained by finding the minimum value of the function defined as follows:

$$MAD(dx, dy) = \frac{1}{64} \sum_{\substack{\forall P_N(i,j) \\ =1}} |C(i, j) - G(i + dx, j + dy)| \tag{3}$$

where $G(i, j)$ represents the MB from a reference picture, and (dx, dy) is a vector representing the search location. From (3), the total number of points used for calculating the MAD is reduced. For example, in each search location using conventional methods, 256 "minus" operations, 255 "sum" operations, and 256 "absolute" operations are required. With our approach, it requires only 64 "minus" operations, 63 "sum" operations, and 64 "absolute" operations. For the motion estimation and compensation of a RMB, the contents of the static region are directly copied from the previous frame, while the motion vector of the moving region is determined by using (3). The determination of the best-matched pattern in a video sequence and the principle of interframe coding for RMB are illustrated in Figs. 5 and 6,

TABLE II
SIMULATION RESULTS

Sequence	Picture format	Target bit rates	Average bit per frame		Average PSNR (dB)	
			H.263	Proposed method	H.263	Proposed method
Akiyo	QCIF	8kbps	735.67	734.73	35.15	35.39
Salesman	QCIF	8kbps	844.81	842.05	31.27	31.35
Akiyo	QCIF	10kbps	914.02	913.11	36.80	36.99
Claire	QCIF	10kbps	940.65	935.75	35.83	36.07
Salesman	QCIF	10kbps	1020.90	1020.86	32.42	32.46
Akiyo	CIF	16kbps	1508.11	1510.84	33.02	33.19
Miss America	QCIF	16kbps	1474.58	1463.46	38.42	38.53
Mother-daughter	QCIF	16kbps	1518.28	1515.08	32.60	32.68
Akiyo	CIF	24kbps	2216.69	2211.39	35.75	35.82
Mother-daughter	QCIF	24kbps	2313.95	2310.36	35.45	35.49
Foreman	QCIF	24kbps	2372.27	2409.95	30.59	30.57
News	CIF	32kbps	3183.74	3198.74	30.25	30.44
Foreman	QCIF	48kbps	4700.82	4730.65	34.50	34.45
News	CIF	48kbps	4805.11	4801.83	33.55	33.58

respectively. As a result, both the prediction errors and the complexity required for motion estimation are reduced.

D. Prediction Error Encoding

The encoding of residual errors for an AMB is based on the conventional DCT-based coding methods. However, if this conventional method is applied directly for encoding a RMB, two blocks will have to be processed. In our approach, the prediction errors of the moving region in a RMB are rearranged to a block of size 8×8, and DCT transform is then employed. Based on the distributions of the residual errors, the rearrangement method is devised without a significant increase in the high order transform coefficients. As a result, only a block is needed to encode for a RMB, and the compression ratio is increased. The rearrangements of the residual errors in the spatial domain for four of the patterns are illustrated in Fig. 7. The arrows in the diagrams indicate the rearrangement order in a MB. The rearrangements for patterns 5–8 are not illustrated, as they are already in the form of a block.

In our coding method, two extra codewords, "MPMODE" and "PMB", are needed. The "MPMODE" uses 1 bit to represent the prediction mode: the conventional block-based prediction or the moving region prediction. The "PMB" uses 3 bits to indicate the pattern being selected for the moving region prediction.

III. SIMULATION RESULTS

The performance of the proposed algorithm is evaluated based on a variety of image sequences, which include typical head-and-shoulders video sequences, smooth motion sequences, and active motion sequences. The results of our proposed method are compared to the H.263. Full-search motion estimation and the TMN-8 [16] rate control method are employed for obtaining the encoding results. Table II and Fig. 8 illustrate the encoding results of the first 100 frames using our approach, as well as the H.263 scheme. It can be observed that our proposed method generally gives better image quality than that of the H.263 for low and high bit rates. However, its performance regarding motion-intensive video (e.g., Foreman) is reduced. There is no prediction gain for such sequences using our approach. This is due to the fact that these kinds of video sequences result in a small number of RMBs,

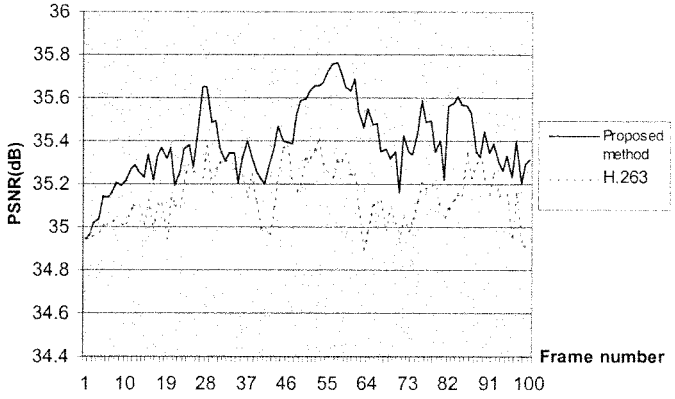

Fig. 8. PSNR of the first 100 frames for the video sequence "Akiyo" (target bit-rate=8 kbps).

TABLE III
ENCODING TIME SAVED PER FRAME COMPARED TO H.263

Sequence	Picture format	Target bit rates	Total number of RMB in the first 100 frames	Encoding time saved per frame
Akiyo	QCIF	8kbps	1803	18.01%
Salesman	QCIF	8kbps	4359	42.31%
Claire	QCIF	10kbps	3226	53.53%
Miss America	QCIF	16kbps	3545	38.71%
Mother-daughter	QCIF	16kbps	1976	8.69%
Akiyo	CIF	24kbps	6009	32.13%
News	CIF	32kbps	9669	34.07%
Foreman	QCIF	48kbps	737	-2.69%

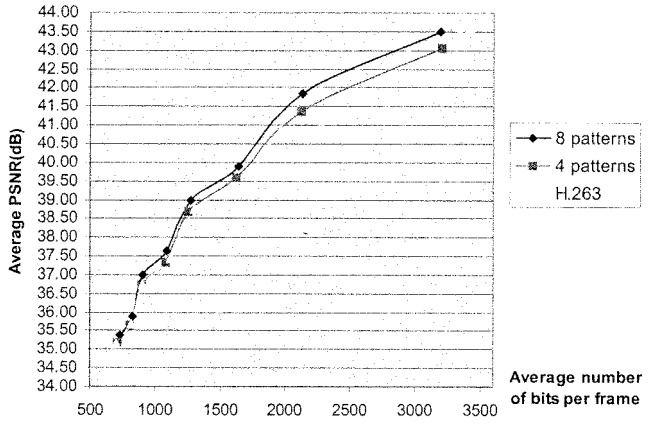

Fig. 9. Simulation results for the video sequence "Akiyo" with different bit-rates.

while our approach requires additional bits to represent the MB type. Consequently, there is no advantage of encoding high motion-intensive sequences using our approach. Table III shows the improvement in the encoding time and the total number of detected RMBs in the first 100 frames. Experiment results show that the encoding time of our approach is much less than that of the H.263. The amount of time saved for encoding a frame for head-and-shoulders or smooth motion sequences varies from 8.69% to 53.53%. For encoding the active sequence "Foreman", the approximating simulation time is 2.69% longer than that using the H.263. As a small number of RMBs are detected in this sequence, the time saved on motion estimation is insufficient to compensate for the time spent on the pre-processing step in our approach. In general, when the number of RMBs detected increases, the required encoding time will also decrease, while the PSNR will increase. However, the run time will become longer and the PSNR will be reduced slightly when fewer RMBs are detected.

We have also selected four pre-defined patterns only, patterns 1 to 4, to encode the sequence "Akiyo." The experiment results are shown in Fig. 9. It can be observed that the PSNR using eight patterns is higher than that of using four patterns and the H.263 scheme at the same bit-rates; the maximum prediction gains using eight patterns compared to the H.263 scheme and the four-pattern approach are about 0.6 and 0.5 dB, respectively. However, the H.263 scheme outperforms the four-pattern approach at high bit-rates. This is due to the fact that the four patterns are insufficient to approximate the moving regions, so the prediction gain is not as good as when using eight patterns or the H.263. In conclusion, the coding efficient using eight patterns is better than that of using four patterns as well as the H.263 scheme.

IV. CONCLUSION

We proposed an efficient very low bit-rate video coding scheme. Eight pre-defined patterns were chosen by experiments to represent moving regions. Based on the predefined patterns, the computation required for motion estimation can be reduced and better prediction gains can be achieved. The encoding time of our approach is much less than that for H.263 for smooth motion sequences. Furthermore, in order to reduce the size of a MB to be encoded, we devised a rearrangement method to compact the residual errors of a MB into a block of size 8×8. The simulation results show that our approach outperforms the H.263 scheme for encoding the head-and-shoulders and smooth motion sequences. However, the total number of detected RMB will decrease in motion-intensive video sequences, in which its performance will be degraded and close to that of the H.263. In conclusion, this approach outperforms the H.263 in terms of the PSNR and the run-time for sequences of smooth motion.

REFERENCES

[1] ITU Telecom. Standardization Sector of ITU, "Video coding for low bitrate communication," Draft ITU-T Recommendation H.263, 1996.
[2] ITU Telecom. Standardization Sector of ITU, "Video coding for low bitrate communication," Draft ITU-T Recommendation H.263 Version 2, 1998.
[3] ITU Telecom. Standardization Sector of ITU, "Video codec for audio-visual services at.$p \times 64$kbit/s," ITU-T Recommendation H.261, 1993.
[4] Y. L. Chan and W. C. Siu, "New adaptive pixel decimation for block motion vector estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 113–118, Feb. 1996.
[5] L. M. Po and W. C. Ma, "A novel four-step search algorithm for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 313–317, June 1996.
[6] C. H. Lin and J. L. Wu, "A lightweight genetic block-matching algorithm for video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 386–392, Feb. 1998.

[7] H.-S. Wang and R. M. Mersereau, "Fast algorithms for the estimation of motion vectors," *IEEE Trans. Image Processing*, vol. 8, pp. 435–438, Mar. 1999.

[8] J. C. Woods, S. Ramanan, and D. E. Pearson, "Low level switched model-based coder for low bit-rate applications," in *Proc. 7th Int. Conf. Image Processing and Its Applications*, vol. 2, 1999, pp. 605–609.

[9] M. F. Chowdhury, A. F. Clark, A. C. Downton, E. Morimatsu, and D. E. Pearson, "A switched model-based coder for video signals," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, pp. 216–227, June 1994.

[10] J. C. Woods, S. Ramanan, and D. E. Pearson, "Selective macroblock level switching for model-based coded video," *Electron. Lett.*, vol. 35, no. 4, pp. 388–389, Mar. 1999.

[11] Y. Nakaya, Y. C. Chuah, and H. Harashima, "Model-based/waveform hybrid coding for videotelephone images," in *Proc. Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 4, 1991, pp. 2741–2744.

[12] 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, pp. 212–220, Feb. 1997.

[13] P. Maragos, "Tutorial on advances in morphological image processing and analysis," *Opt. Eng.*, vol. 26, no. 7, pp. 623–632, 1987.

[14] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Reading, MA: Addison-Wesley, 1992.

[15] Y.-L. Chan and W.-C. Siu, "Block motion vector estimation using edge matching: An approach with better frame quality as compared to full search algorithm," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 2, 1997, pp. 1145–1148.

[16] ITU Telecom. Standardization Sector of ITU, "Video codec test model, near-term, version 8 (TMN8)," Video Coding Experts Group, June 1997.