IMPROVED SCHEMES FOR INTER-FRAME CODING IN THE H.264/AVC STANDARD

Andy C. Yu, Graham R. Martin and Heechan Park

Department of Computer Science, University of Warwick, Coventry, CV4 7AL, United Kingdom
Email: {andycyu, grm, heechan}@dcs.warwick.ac.uk

ABSTRACT

An efficient algorithm for inter-frame coding in the H.264/AVC standard is extended to provide more significant speedup in computational performance for sequences containing high spatial correlation and motion. The proposed scheme features a more sophisticated search process and robust predictions to achieve better PSNR-rate performance for a large range of compression levels. Extensive simulation results demonstrate speedups of between 41% and 68%, with no noticeable deterioration in picture quality or compression ratio, even for the coding of complex video sequences.

1. INTRODUCTION

Block-matching motion compensation is a core part of inter-frame coding in the H.264/AVC standard. It aims to reduce the coding of redundant information by identifying the best matching pixel data in adjacent frames. The difference with other standards, however, is that the block size is no longer fixed, but ranges from 4×4 to 16×16 pixels. In order to choose the best block size for a macroblock, the H.264 standard makes use of computationally intensive Lagrangian rate-distortion (RD) optimization [4], the general equation of which is:

$$LC\,(\text{Qp}) = D(\text{Qp}) + \lambda_{\text{mode}}.R(\text{Qp}) \tag{1}$$

where LC(Qp) is the rate-distortion (RD) cost. The Lagrangian multiplier, λ_{mode}, is associated with the quantisation factor, Qp, and has the relationship:

$$\lambda_{\text{mode}} = 0.85 \times 2^{(\text{Qp}-12)/3} \tag{2}$$

D(Qp) is a measure of the distortion between the original macroblock and the reconstructed macroblock located in the previous coded frame, and R(Qp) reflects the number of bits, including the macroblock header, motion vectors and all the DCT residue blocks. Both D(Qp) and R(Qp) are associated with the chosen mode and the quantisation factor, Qp.

In inter-frame coding, possible inter-modes are:

$$\{\text{SKIP, INTRA}, 16\times16, 16\times8, 8\times16, \text{P}8\times8\}$$

where SKIP is a direct copy from the previous frame; INTRA represents intra-modes predicted from encoded adjacent pixels, and P8x8 accounts for the inter modes with small partition size (8x8, 4x8, 8x4, and 4x4 pixels). The optimal mode for a macroblock is selected as that which produces the least RD cost. Since the exhaustive search method is employed in all the inter-modes to acquire a final mode decision, the computational burden of the search process is far more significant than any existing video coding algorithm.

In previous work, a fast algorithm, MFinterms, [1] has been proposed to alleviate inter-frame encoder complexity, while still maintaining picture quality and coding efficiency at certain compression levels. Success of the MFinterms algorithm is achieved by discriminating two kinds of encoded macroblocks: (a) macroblocks encoded with SKIP mode; (b) macroblocks encoded by inter-modes with the larger decomposed partition sizes (greater than 8x8 pixels). The MFInterms algorithm assigns different categories of inter-modes to each macroblock with a complexity measurement defined as follows:

$$\text{Complexity} = \begin{cases} \text{high}, & \ln(\text{E}_{\text{AC}})/15.25 > \text{Th}_{\text{c}} \\ \text{low}, & \ln(\text{E}_{\text{AC}})/15.25 < \text{Th}_{\text{c}} \end{cases} \tag{3}$$

where Th_c is a spatial complexity threshold and E_{AC} is the total energy of the high-frequency component (AC component) in the current macroblock. If the result in (3) is low, the algorithm checks those inter modes with partition sizes of larger than 8x8. Otherwise, the macroblock is decomposed into four 8-by-8 blocks and a measurement of block-based motion consistency is made to determine whether inter-modes with a smaller partition size are required.

The MFInterms algorithm was shown to achieve a significant reduction in computation time [1]. However, the performance gains were dependent on the spatial content and extent of motion in the test sequences, for a limited range of compression levels. In this paper, we propose improved schemes for the MFinterms algorithm which achieve a better PSNR-rate performance and a reduced computational requirement for any coding condition. The remainder of this paper is organized as follows. In Section 2, the improved schemes for the MFinterms algorithm are presented. Extensive simulation results in terms of PSNR-rate performance and algorithm efficiency are presented in Section 3 and conclusions are drawn in the Section 4.

2. THE PROPOSED SCHEMES FOR THE MFINTERMS ALGORITHM

The proposed scheme contains a more sophisticated search process and robust predictions in selecting an optimal inter-mode for each macroblock. The scheme is described as comprising three levels. Each level targets a different category of inter-modes according to the complexity of the search processes. The following subsections introduce these three levels.

3.1. The first level of the proposed algorithm

The first level of the proposed algorithm distinguishes the macroblocks encoded with SKIP (denoted as skipped macroblocks). Fig. 1 illustrates the detailed flowchart for this level. The skipped macroblocks can be found in P-frames where the pixel information is almost identical to that in the corresponding same position in the previous frame. Thus, they can be detected by means of the temporal similarity between the two macroblocks. As skipped macroblocks tend to occur in clusters, such as in a patch of static background, we suggest that the current macroblock undergoes temporal similarity detection if at least one of two possible valid skipped macroblocks is found above or to the left of the current macroblock. The decision for the temporal similarity detection is defined as

$$\text{Decision} = \left\lfloor \frac{x_{SAD}}{\text{SAD}_{\text{current}}(t;t-1)} + \varepsilon \right\rfloor \tag{4}$$

where $\text{SAD}_{\text{current}}(t;t-1)$ is the sum absolute difference (SAD) of the macroblocks in frames t and $t-1$; ε represents a tolerance constant; x_{SAD} is the SAD of the available skipped neighbour. An average of the SAD values is computed if both skipped neighbours are valid. A non-zero outcome in (4) indicates that the current macroblock is encoded with SKIP only. Otherwise, further examinations in the second level are required. The algorithm for this level is summarised as follows:

1.1. Deactivate all inter-modes.

1.2. Determine whether one of the encoded neighbours located above or to the left is a skipped macroblock. Execute 1.3 if the situation is true. Otherwise, proceed to next level.

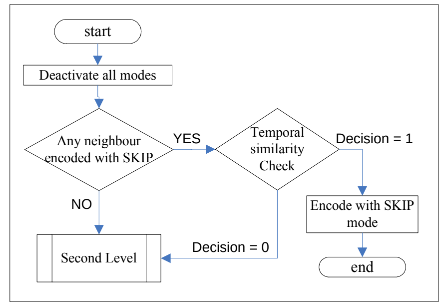

Fig. 1 Flowchart of 1st level of improved scheme.

1.3. Perform the temporal similarity test described in (4). If a non-zero decision is obtained, encode the current macroblock with SKIP mode and record its SAD value. Otherwise, further examinations in the next level are required.

3.2. The second level of the proposed scheme

Fig. 2 illustrates the flowchart of the second level. This level targets those macroblocks encoded with inter modes with a large partition size (8×16, 16×8, and 16×16 pixels). The general tendency that inter modes with large partition sizes are more suitable for the encoding of homogeneous content has been verified by a number of authors [1,2]. The reasons given are that: (a) homogeneous macroblocks tend to contain fewer moving features requiring multiple motion descriptors, and (b) owing to the homogeneous content, the distortion costs arising from incorrect predictions are often insignificant.

The spatial complexity measurement in (3) is employed to determine if the current macroblock requires further examination by inter-modes with smaller partition size. However, (3) excludes the case of macroblocks with spatial content in between the two-tiered classifications. Furthermore, it is observed that the mode decision for the aforementioned macroblocks varies according to what level of compression is applied. Since the mode decision for a macroblock is determined by the lowest RD cost, we suggest computing the RD cost of a few inter-modes before the entire search process is performed. If the best mode for a high-detailed macroblock in this level is in favour of using inter-modes with partition size of 16x8 and 8x16, a more thorough search in the next level is required. Otherwise, the mode decision for the current macroblock is made. The detailed algorithm at this level is described as follows:

2.1. Determine the complexity of the macroblock using (3). If the content of the current macroblock is homogeneous, select the best mode for the current macroblock either from SKIP or the inter-mode with partition size of 16×16. Otherwise, continue to 2.2.

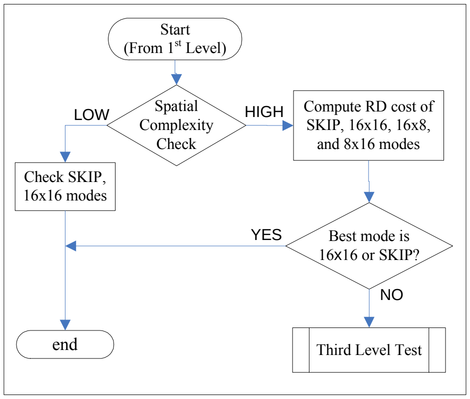

Fig. 2 Flowchart of 2nd level of improved scheme.

2.2. Activate the inter-modes with partition size of 8×16, 16×8, and 16×16 and SKIP. Check the current macroblock with the activated inter-modes.
2.3. Compute the RD cost of the four modes and obtain a mode decision for this level. If the optimal mode is not SKIP or 16×16, proceed to the third level. Otherwise, the mode decision for the current macroblock is decided.

3.2. The third level of the improved scheme

The third level computes searches within the P8x8 mode (including the inter-modes with smaller partition sizes of 8×8, 8×4, 4×8, and 4×4 pixels). Since each decomposed 8-by-8 block has to undergo search operations, a very large computational overhead is expected. Fig. 3 illustrates the proposed scheme for this level. The current macroblock is decomposed into four non-overlapping 8×8 blocks. Each block is then checked by the inter mode with partition size of 8×8. The other inter-modes with smaller partition sizes are activated if the following condition is true

$$LC_{8\times8}(N^{\text{th}}) < LC'/4 \qquad (5)$$

where $LC_{8x8}(N^{th})$ is the RD cost of the N^{th} current 8×8 block and LC' represents the RD cost of the best mode obtained from the last level. Note that (5) does not need to be revised if all inter-modes are activated. Thus, a computational saving is achieved for the first few blocks which do not satisfy the condition in (5). The detailed algorithm of this level is described as follows:

3.1. Decompose the current macroblock into four non-overlapping 8×8 blocks.
3.2. Activate on the inter mode with partition size of 8×8 only.
3.3. Check and compute RD cost for the activated inter-modes for the current 8×8 block until all the blocks have been examined.

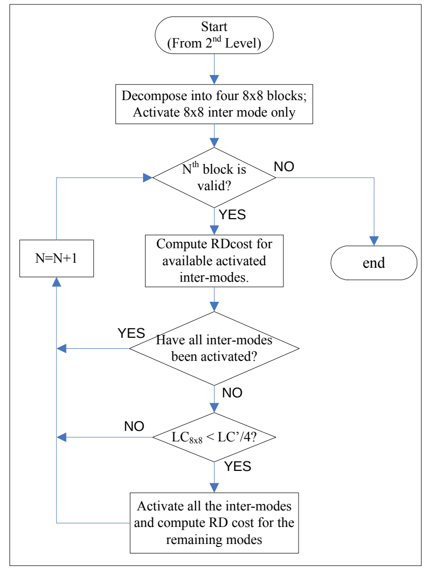

Fig. 3 Flowchart of the 3rd level of improved scheme.

3.4. Detect if the other available inter-modes have been activated. If the situation is true, proceed to 3.3. Otherwise, examine the condition (5) and implement 3.5.
3.5. If the condition in (5) is satisfied, activate all the available inter-modes in this level. Otherwise, return to 3.3.

4. SIMULATION RESULTS

This section compares the results of the proposed algorithm incorporating the improved schemes with the previously reported MFInterms algorithm [1]. Results are presented as improvements over the standard H.264 benchmark, software version JM6.1e[3]. Except were stated, the selected sequences are of QCIF resolution (176×144 pixels). The other settings are as follows: in total 100 frames of each sequence were processed. The frame rate and GOP are 30 frames/sec and 10 frames, respectively. The precision and search range of the motion estimation is set to ¼ pixel and ±8 pixels, and finally, CABAC coding is utilised.

Fig. 4 exhibits two PSNR-rate relationship diagrams for the Flower sequence in CIF resolution (top diagram) and the Stefan sequence in QCIF resolution (bottom). Both sequences are classified as Class C sequences. It is clear that the PSNR-rate performance of the new algorithm is virtually identical to that provided by the JM6.1e software, at all bit rates. However the previously reported MFinterms algorithm shows a decreased PSNR, particularly at the higher bit rates. Consequently we conclude that the new algorithm provides better performance than the MFinterms algorithm at all bit rates.

Table 1 summarises the speed up over the JM6.1e benchmark for the two algorithms. Qp was fixed at 32. Generally, both algorithms provide different degrees of speed up depending on the class of the selected test sequences. Both algorithms reduce the computation time for Class A and Class B sequences by in excess of 57%. For Class C sequences (that generally are more difficult to encode), the new scheme performs almost twice as well as the previously reported MFinterms algorithm, providing speedups of 41%-55% compared with 23%-30%.

Fig. 5 illustrates the speed up achieved by both algorithms for different Qp settings, for the Stefan sequence. Significantly, the proposed algorithm maintains the same speed up of around 42% for low Qp values (high bit rates) and then increases to over 60%; while a slow increase of between 17% and 34% is shown for the MFinterms algorithm. Fig. 5 illustrates that the proposed algorithm is superior to MFinterms at all bit rates.

Class	Sequence	MFinterms [1]	Proposed scheme
A	Container Ship	72.94%	68.10%
B	Silent Voice	60.22%	57.61%
C	Car Phone	30.97%	51.60%
	City	22.85%	54.74%
	Crew	24.39%	51.76%
	Flower (CIF)	27.54%	48.40%
	Ice	29.91%	44.10%
	Stefan	28.72%	41.45%
	Tempete (CIF)	20.87%	44.37%

Table 1 Computational Speedup compared to JM6.1e

5. CONCLUSIONS

Improved schemes for block size selection in inter-frame coding in the H.264/AVC standard have been proposed. The new algorithm significantly improves the PSNR-rate performance of the previously reported MFinterms algorithm at all bit rates, while almost halving the computational requirement for Class C sequences. The simulation results further show that the proposed algorithm achieves the same coding performance in terms of picture quality and compression ratio as that of the H.264/AVC standard, yet reduces the computational requirement by up to 68%.

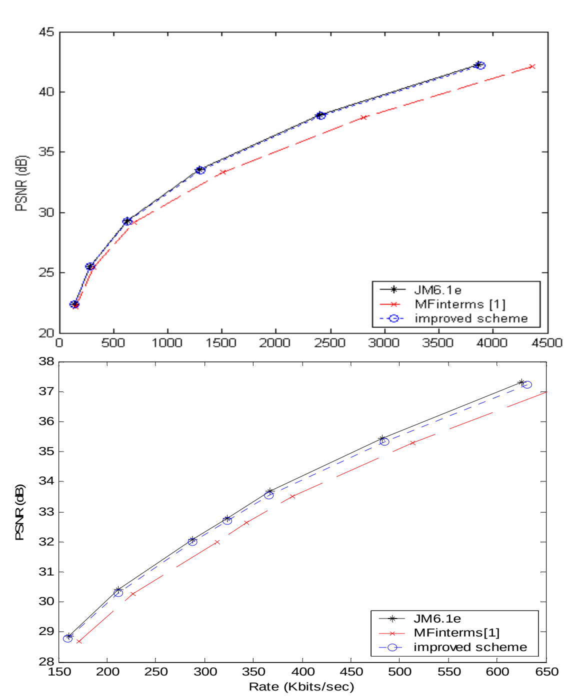

Fig. 4 PSNR-rate relationship diagrams for Flower in CIF (top) and Stefan in QCIF (bottom) resolutions.

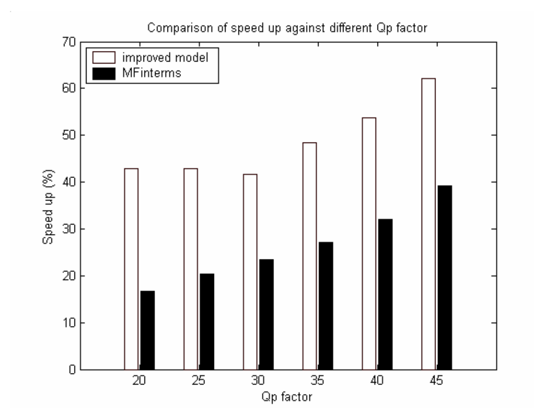

Fig. 5 Speedup for Stefan (QCIF) sequence for different Qp values.

6. REFERENCES

[1] A. Yu and G. Martin, "Advanced block size selection algorithm for inter-frame coding in H.264/AVC," *Proc. ICIP 2004*, Singapore, pp. 95-98, Oct. 2004.

[2] I. Richardson, "H.264/MPEG-4 video compression: video coding for next-generation multimedia," Wiley, London, 2003.

[3] JVT reference software for the H.264/AVC standard, JM 6.1e, downloaded from http://bs.hhi.de/~suehring/tml.

[4] ISO/IEC 14496-10:2003, "Information technology – coding of audio visual objects – Part 10: advanced video coding," Dec. 2003.