

A NEW HYBRID BLOCK-MATCHING MOTION ESTIMATION ALGORITHM

Jun Ge and Gagan Mirchandani

Department of Electrical and Computer Engineering
The University of Vermont
Burlington, Vermont 05405

ABSTRACT

A fast and efficient hybrid block-matching motion estimation algorithm is proposed in this paper. The fact that motion characteristics are relatively stable in a GOP (group of pictures) is utilized to reduce the computation burden without quality degradation. For P-frames, a multi-resolution block-matching algorithm is used. At the coarsest level, edge information is used to reduce searching positions before an exhaustive search. At the finer levels, only the several candidates chosen at the upper level are verified. For B-frames, *mean-absolute-error* (MAE) is used not only for the matching criteria, but also for classifying the macroblocks with the motion information extracted from P-frames. Concurrently, the search window size is adjusted using the contribution of different motion range to the improvement of the picture quality with motion compensation. When the decoded I-frame and P-frames are used as the reference, experimental results show that reconstructed digital frames have even better quality than the full search algorithm in terms of PSNR, under MPEG-1 coding environment.

1. INTRODUCTION

Motion estimation/compensation is the standard approach to reducing temporal redundancy in coding of real-world video sequences. Due to its simplicity, the block-matching algorithm (BMA) has been widely used in motion estimation. If exhaustive search (ES) is used in a BMA, the complexity of the video coder becomes dominated by the cost of motion estimation. ES exhaustively searches for the best matched block within a predefined size search window to get the optimal solution in terms of the quality of motion compensation. Quality is usually measured with the MAE, or *sum-of-absolute-difference* (SAD) for each macroblock. Since massive computation is required in the implementation of ES, many fast search algorithms [3]-[7] have been developed to reduce the computation burden.

Motion estimation algorithms are focused on the speed of finding optimum displacement vectors that generate the

least degradation in image quality. The following issues are often considered: (1) In many video compression standards, the image is divided into macroblocks (typically 16x16 pels). A macroblock consists of four blocks (8x8 pels) of luminance and one block for each of the two chroma components. The motion vector is searched in a macroblock basis but the compression of the motion compensated macroblock is performed by using DCT on a block basis. Therefore, the optimal motion vector does not necessarily have a high compression performance in block-based DCT coding [3]. In other words, a sub-optimal motion vector may have the same *peak signal-to-noise ratio* (PSNR), but higher compression performance and require less computation time. (2) Many fast search algorithms do not use the motion information within a GOP to guide the motion estimation. Experimental results [1] show that the motion vector distribution of real-world video sequences are highly center-biased. The motion information within a GOP may be exploited to adapt the search window size and classify the macroblocks into different types. (3) For real-world video sequences, motion vectors are generally correlated spatially and temporally. Those correlations may be used to reduce the search space [5]. They may also help in reducing the bit rate for motion vectors, since standards like H.263 [11] use differentially coded motion vectors in each row of macroblocks in a group-of-blocks. (4) When using a multiresolution analysis, only dc values are used at coarse scales.

We propose a new hybrid motion estimation algorithm that takes into account the above four issues, except that we use edge orientation information at the lowest scale in our multiresolution analysis. In Section 2, we describe the techniques used in the new algorithm. Section 3 gives the proposed algorithm. Simulation results are presented in Section 4 and conclusions in Section 5.

2. THE TECHNIQUES

2.1. Modified Exhaustive Search Algorithm

Most fast block-matching algorithms restrict the number of search locations presuming a unimodal error surface model,

i.e., the matching error increases monotonically as the search moves away from the position of the global minimum error. Because this assumption is not always true for real-world video sequences, a local optimal is generally found. The *successive elimination algorithm* (SEA) [7] drastically reduces the computation load of the motion estimation without losing optimality of the motion vector field. In contrast to other fast algorithms, the SEA excludes many search positions in calculation of the error, by evaluating lower bounds for the error measure. In [6], the SEA is improved by introducing tighter bounds for the SAD and by exploiting the already calculated lower bounds during the calculation of the matching criterion.

In the proposed algorithm, a multi-resolution block matching algorithm, similar to [5] is used to perform the motion estimation for the two P-frames in a GOP. We first take the *Wreath product transform* (WPT [8], [9]) to obtain the multiresolution representation of the current frame. At the coarsest level, a full search which uses the technique in SEA is performed by using not only the MAE but also the edge orientation information [10]. The reason for using edge orientation is that the difference between edge orientations for unmatching macroblocks will be relatively larger than that of matching macroblocks. Some unmatching macroblocks will be picked out by comparing the edge orientations in advance. Concurrently, the spatial correlation of motion vectors is also used for searching continuous motions. After the coarsest level search, several initial motion vector candidates are chosen for the next level. At the following levels, the motion vector candidates are refined within the smaller search window as in [5].

2.2. Motion Information to Guide the Estimation

2.2.1. Classification of Macroblocks

Although motion estimation is not normative in all standards, the formation of the end result of the motion estimation is specified [2]. When selecting the macroblock type for B-frames, the encoder needs to calculate the best motion compensated macroblock for forward motion compensation. It then calculates the best motion compensated macroblock for backward motion compensation by a similar method. Finally it averages the two motion-compensated macroblocks to produce the interpolated macroblock. It then selects the one with the best performance. After this step, if the motion compensated macroblock is only slightly better than the uncompensated macroblock, then the zero vector is assigned to this macroblock. For those macroblocks with motion vector (0, 0), the least number of bits are allocated to the motion vector, if *variable length coding* (VLC) is used to code the motion vectors. Those macroblocks are coded in intra-mode. We show in Table 1 (UMB stands for uncompensable macroblocks) the average number of un-

compensable macroblocks for various test sequences using exhaustive search. On average, about half of the macroblocks are uncompensable. If the uncompensable macroblocks can be classified before the motion compensation, the computation burden can be reduced further. For I-frame and P-frames, MPEG [2] suggests using more bits to guarantee the quality of the pictures. Therefore in the proposed algorithm, only the GOP motion information is used for the classification of uncompensable macroblocks for B-frames. By using this technique, the computation burden is seen to be reduced further by almost a half.

Sequence	Num of UMBs	Percentage
tennis (CIF)	217/(18x22)	72
missa (CIF)	177/(18x22)	81
salesman (CIF)	254/(18x22)	68

Table 1. Percent of uncompensable macroblocks

The uncompensable macroblocks could be : (a) those corresponding to a stationary background; (b) those corresponding to an uncovered background and new objects where no information can be found from the previous reference frame, or (c) those with nonhomogeneous motion because one vector compensate one or more of its four blocks incorrectly.

In [3] the reversed square sum (RSS) is proposed as the matching criterion to find the uncompensable macroblocks. The frame difference contrast (FCON) and the local variance contrast (LCON) [4] are used to find the macroblocks with nonhomogeneous motion. But the performance of these algorithms strongly depend on the choice of some threshold. Furthermore, the computations are intensive, and the characteristics of the video sequence are not taken into account to guide future estimation.

We propose to use MAE between the macroblock in the current B-frame and the corresponding macroblock in the reference frame with motion vector (0, 0) to find the uncompensable macroblocks according to the following two thresholds:

$$T_i = R_i \times MAE_{avg}^P, i = 1, 2$$

where MAE_{avg}^P is the average macroblock MAE from the motion estimation of two P-frames and R1 and R2 are two predefined ratios. If the MAE_{MB} (MAE for an MB) is smaller than T1, the macroblock is defined as a type-A uncompensable macroblock. If the MAE_{MB} is large than T2, then we compare it with the MAE of the corresponding macroblock in reference frames (past and future frames). If there is no significant improvement, this macroblock is identified as a type-B macroblock. For other macroblocks, we need to classify the blocks in it. If all four blocks are compensable, the macroblock is identified as compensable, otherwise it is classified as a type-C uncompensable mac-

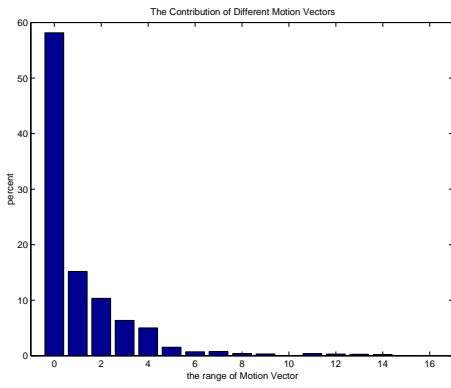


Fig. 1. Average contribution of motion vector size

roblock. The threshold values are dependent on the the average MAE of the two P-frames and two predefined ratios. We assume that the average MAE for the B-frames will not be much larger than the average MAE for the P-frames, except for the occurrence of scene cuts [4].

2.2.2. Adaptive Window Size Search

Macroblocks with different motion vectors contribute differently to the reduction in total redundancy. Fig. 1 shows the average contribution of motion vectors with different sizes for the "salesman" sequence. After the motion estimation for the two P-frames, if only a small number of macroblocks have large motion vectors, and no significant contribution, the search window size may be adjusted to reduce the computation. Therefore, the motion information for the GOP is also used to derive the search window size. By considering VLC coding of the motion vectors, the window sizes are classified as small (window = 4), medium (window = 8) and large (window = 16). The initial window size is large. Window size is adjusted as follows: (1) For motion vectors with $|mvx|$ or $|mvy| > 8$, if the contribution is less than some threshold value, the window size is set at medium. $|mvx|$ and $|mvy|$ are motion displacements in vertical and horizontal directions respectively. (2) For motion vectors with $|mvx|$ or $|mvy| > 4$, if the contribution is less than some threshold value, the window size is set at small.

3. THE PROPOSED MOTION ESTIMATION ALGORITHM

The following is the proposed algorithm. First, the GOP is set as I-B-B-P-B-B-P-B-B-I(next GOP) which are in display order. The I-frame is coded first. Then the modified exhaustive search motion estimation algorithm is performed for the two P-frames. Then, the motion vector fields for the two P-frames and the predicted frames are used to create

the macroblock-type-table and the window size adapted for the group of pictures. After the coding of P-frames, the bi-directional motion estimation is performed for each B-frame. The macroblock-type-table is used to decide whether we need to perform motion estimation for each macroblock. The modified window size helps reduce the computation burden.

For motion estimation for P-frames, the three-level multiresolution frame structure and the conventional three-step search method are used [5]. The average speedup factor is about 38 compared to the computation needed for the exhaustive search. It is better than the speedup factor 27 in [5]. For the motion estimation of B-frames, the percentage for the found uncompensable macroblocks represents the reduction in computation. In the classification of macroblocks, the only additional computation is for the comparison, because we do not need to recalculate the matching metric-MAE. Therefore the final speedup factor for B-frames is about 57 on average for the three test sequences: "tennis", "missa" and "salesman". Table 1 also shows the percentage of uncompensable macroblocks found before the motion estimation for B-frames.

4. SIMULATION RESULTS

All the three test sequences are in CIF format (352 x 288). The frame rate is 30 frames/sec. Fixed quantization scale is used in coding. For motion estimation, the decoded past and future frames are used as reference frames. The PSNR is used widely as the performance measure in video compression. For video sequence with pixel depth of 8 bits, the PSNR is defined as

$$10 \log_{10} \frac{255^2}{\frac{1}{N_v N_h} \sum_{i=0}^{(N_v-1)} \sum_{j=0}^{(N_h-1)} [I(i, j) - \hat{I}(i, j)]^2}$$

where $I(i, j)$ denotes the original frame and $\hat{I}(i, j)$ is the reconstructed frame (being rounded towards the nearest integer). N_v and N_h are the numbers of pels in vertical and horizontal directions respectively.

The average bit-rate and PSNR of the compressed sequence were analyzed using three different block-matching algorithms: the full-search algorithm, the MRBMA algorithm [5], and the proposed algorithm. Fig. 2 shows the PSNR comparisons with the same average bit-rates, for the first 36 frame of the "missa" sequence using different algorithms. The algorithms obtain almost similar performance. However, the proposed algorithm gives a better average PSNR for B-frames by almost 2dB. That is shown in Fig. 3 where we see the average bit-rate versus the average PSNR of "missa" sequence for the three algorithms. Table 2 shows the average PSNR for the first 36 frames (four GOPs) of three video sequences.

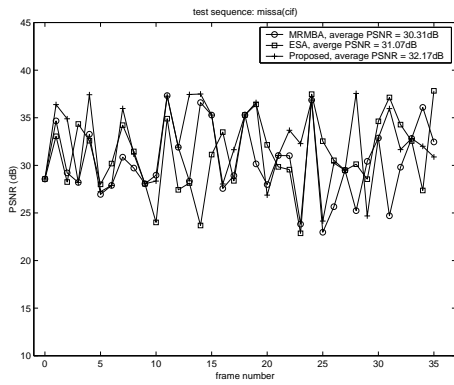


Fig. 2. Comparison of PSNR for "missa"

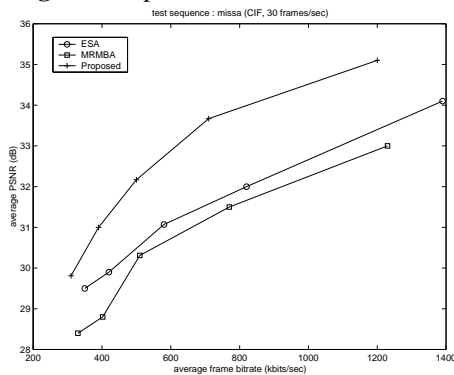


Fig. 3. Average bitrate versus the average PSNR

Algorithm	salesman	tennis	missa
ESA	31.98	33.13	31.07
MRBMA	31.43	33.01	30.31
Proposed	32.22	33.14	32.17

Table 2. Average PSNR(dB) for the first 36 frames of three video sequences

5. CONCLUSION

In this paper, a new hybrid motion estimation algorithm which consider the "speed-quality-bitrate" tradeoff was proposed. The fact that motion characteristics are relatively stable in a GOP is utilized to reduce the computation burden without quality degradation. For P-frames, since edge orientation is used in the coarsest level of the multi-resolution block matching algorithm, the computation burden is reduced by about 30 percent comparing to MRBMA. For B-frames, MAE is used not only for the matching criteria, but also for classifying the macroblocks with the motion information extracted from P-frames. Therefore, almost no overhead computation is needed for classifying the uncompensable macroblocks. When the decoded I-frame and P-frames are used as the reference, experimental results show that reconstructed digital frames have even better quality than the

full search algorithm in terms of PSNR.

6. REFERENCES

- [1] 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, no. 3, pp. 313-317, June 1996.
- [2] ISO/IEC JTS 11172-2, "Information technology – Coding of moving pictures and associated audio for storage at up to about 1.5 Mbit/s, Part2: Video" Nov. 1993.
- [3] S.S. Kim, J.S. Kim, and H.W. Park, "Subblock sum matching algorithm for block-based interframe coding," *Signal Processing: Image Communication* 15 (1999) pp. 189-199.
- [4] J.-P. Fan, and F.-X. Gan, "Motion Estimation Based on Global and Local Uncompensability Analysis," *IEEE Trans. Image Proc.* vol. 6, no. 11, Nov. 1997.
- [5] B.C. Song and J.B. Ra, "A fast multi-resolution block matching algorithm for motion estimation", *Signal Processing: Image Communication* 15 (2000) pp. 799-810.
- [6] M. Brunig and W. Niehsen, "Fast full-search block matching", *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 2, pp. 241-247, Feb. 2001
- [7] W. Li and E. Salari, "Successive elimination algorithm for motion estimation." *IEEE Trans. Image Processing*, vol. 6, pp. 1587-1591, Nov. 1997.
- [8] R. Foote, G. Mirchandani, D. Rockmore, D. Healy and T. Olson, "A wreath product group approach to signal and image processing: part I – multiresolution analysis," *IEEE Trans. Signal Processing*, Jan. 2000. vol. 48, no. 1, pp 102-132.
- [9] G. Mirchandani, R. Foote, D. Rockmore, D. Healy and T. Olson, "A wreath product group approach to signal and image processing: part II – convolution, correlation, and applications," *IEEE Trans. Signal Processing*, March 2000, vol. 48, no. 3, pp. 749-767.
- [10] V. Chickanosky and G. Mirchandani, "Wreath products for edge detection", *Proceedings ICASSP 1998*, vol. 5, May 1998, pp. 2953-56.
- [11] "Draft Recommendation H.263: Video Coding for Low Bitrate Communication. ITU-T(CCITT), Dec 1995.