Low complexity motion estimation utilising spatial correlation

S.-I. Park and I.-C. Park

An efficient algorithm is proposed to reduce the computational complexity of block matching motion estimation by using the characteristics of spatial correlation. The proposed algorithm is to skip the motion vector search of inside macroblocks surrounded by identical motion vectors. Experimental results show that the proposed algorithm reduces computational complexity by 52.5% compared to conventional motion estimation at the cost of negligible performance degradation.

Introduction: To reduce the temporal redundancy of video sequences, block matching motion estimation has been adopted in various video coding standards such as ISO/IEC MPEG-1, MPEG-2, MPEG-4 and ITU-TH.263 [1]. In those applications, motion estimation has a large effect on overall computational complexity. Many algorithms have been developed to reduce the search space of motion vectors [2]. Based on the fact that there is little motion difference among spatially neighbouring macroblocks, the neighbouring motion vectors are utilised to reduce the candidate locations to be searched for a macroblock [3] or to approximate the motion vector without exploiting the search space [4]. In [4], the motion vector of a macroblock is approximated by the median value of three motion vectors of the left, upper, and upper right macroblocks if the resulting distortion is not larger than the median distortion of the neighbouring macroblocks. Such a block-by-block approximation has a limitation in reducing the computational complexity although it reflects the motion behaviour of macroblocks.

In this Letter, we propose a low complexity motion estimation algorithm that uses the spatial correlation to further skip the motion vector search by conditionally identifying a group of macroblocks being moved together. The proposed algorithm reduces computational complexity while maintaining the quality of motion estimation.

Proposed algorithm: Motion vectors in an image frame are smoothly varying and discontinuous only at the boundaries of objects moving in different directions or with different velocities, which means that a considerable number of contiguous macroblocks have identical motion vectors in real image sequences. If the boundary of the contiguous macroblocks having identical motion vectors can be identified, we can skip the motion vector search of the inside macroblocks. Although the inside macroblocks surrounded by identical motion vectors do not always have the same motion vectors, the error caused by the assumption that all the inside macroblocks have identical motion vectors may be small owing to the spatial correlation if the distance is not far from the boundary. The problem is how to detect the boundary. To approximate the boundary in a row, we search the left (B_L), upper (B_U), and right (B_R) macroblocks that have identical motion vectors, as shown in Fig. 1, where the number of inside macroblocks is denoted as N.

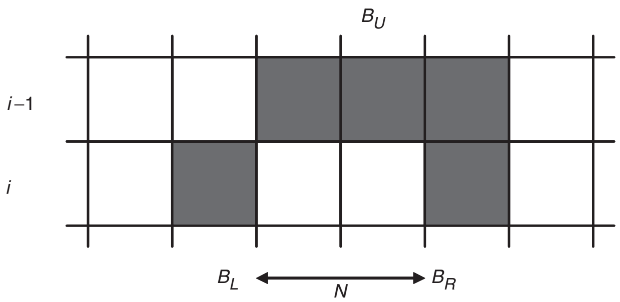

Fig. 1 *An ith row boundary composed of grey-coloured macroblocks with identical motion vectors*

The proposed algorithm consisting of three steps is illustrated in Fig. 2. First, the motion vector search is performed for the current macroblock in a row. If the motion vector of the current macroblock, MV_L, is identical with that of the upper right macroblock, go to the second step. Otherwise, the first step is repeated for the next macroblocks until B_L is identified. Secondly, the motion vectors for the upper macroblocks are checked forward to find the initial column position of a B_R candidate until a different motion vector is found or N becomes larger than a threshold N_{TH}. The initial column position is kept. Thirdly, it is checked whether the first B_R candidate in the initial column of the current row has the identical motion vector. In that case, the macroblocks between B_L and B_R are assumed to have MV_L. Otherwise, the step is repeated by moving backwards from the initial column position until we find a valid B_R. The backward search increases the computational complexity. The macroblocks whose motion vectors are already computed in the third step are skipped in the next B_L identification.

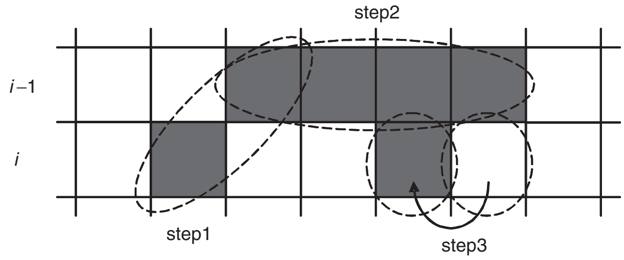

Fig. 2 *Illustration of proposed algorithm*

To reduce further the computational complexity of the third step, a B_R candidate is validated in two stages. The motion compensated error is defined as the sum of absolute differences (SAD) in this Letter. The motion compensated error of the candidate, SAD_R, is calculated with MV_L and compared to that of the B_L, SAD_L. If SAD_R is not larger than SAD_L, the candidate is considered as B_R. Otherwise, the motion vector of the candidate is actually searched. If the found motion vector is equal to MV_L, the candidate becomes B_R.

The number of inside macroblocks affects the computational complexity and the accuracy of motion estimation. As more macroblocks are considered to have identical motion vectors, the computational complexity is further reduced by skipping the search of motion vectors. However, the possibility to include macroblocks with different motion vectors increases and the motion estimation error can be larger. To compromise this conflict, N_{TH} should be carefully decided. The motion vectors of the upper macroblocks play an important role in determining the boundaries of the current row. Since the motion vectors estimated incorrectly in a row can be propagated to the next rows, the resulting distortion can be significant, especially for large N_{TH}. Wherever N reaches N_{TH} in a row, therefore, N_{TH} is temporarily switched to $N_{TH\mathrm{MIN}}$ in the next row to alleviate the propagation problem.

Experimental results: To evaluate the proposed algorithm, 200 frames of seven test sequences represented in the common intermediate format of 352 pels × 288 lines were tested. The motion estimation is carried out on full-pel precision and SAD is used as a matching criterion. The search window range is $[-7, +7]$, and the full search is applied to find the motion vector of a macroblock. By varying N_{TH} from 2 to 8, three versions of the proposed algorithm were evaluated with setting $N_{TH\mathrm{MIN}}$ to 2.

Table 1: Comparison of average PSNR (dB)

Test sequence	CON	MED	Proposed algorithm		
			$N_{TH}=2$	$N_{TH}=4$	$N_{TH}=8$
Coastguard	30.03	29.92	29.84	29.88	29.74
Container	38.46	38.46	38.46	38.46	38.46
Mobile & Cal.	24.34	24.24	24.24	24.19	24.14
Flower	25.97	25.95	25.93	25.92	25.91
Mother	39.52	39.45	39.42	39.41	39.32
News	36.90	36.77	36.69	36.56	36.38
Foreman	31.86	31.75	31.78	31.76	31.71
Average	32.44	32.36	32.34	32.31	32.24

Table 1 compares the average peak signal-to-noise ratios (PSNRs) of the proposed algorithm, the previous MED algorithm [4], and the conventional algorithm (CON) that searches motion vectors for all the macroblocks. In the case of $N_{TH}=2$, the PSNR degradation of the proposed algorithm is almost the same as that of the MED algorithm and less than 0.1 dB even compared to the conventional algorithm.

The backward search rate (BSR), which is defined as the number of the invalid B_R's divided by the total number of B_R candidates, is examined to find the relation between N_{TH} and the number of backward searches. As indicated in Table 2, the BSR increases according to N_{TH}, but is low enough to reduce the computational complexity, especially in Container and News sequences. Table 3 shows the computational complexity normalised by the conventional algorithm. Compared to the MED algorithm, the proposed algorithm with $N_{TH} = 8$ reduces the computational complexity by 24.7% on average.

Table 2: Backward search rate (%)

Test sequence	Proposed algorithm		
	$N_{TH}=2$	$N_{TH}=4$	$N_{TH}=8$
Coastguard	11.5	12.6	15.7
Container	0.2	0.3	0.3
Mobile & Cal.	11.2	13.7	16.6
Flower	17.6	22.3	27.1
Mother	13.8	16.1	17.8
News	5.4	6.7	8.0
Foreman	19.8	22.1	27.4
Average	11.4	13.4	16.1

Table 3: Comparison of average complexity (%)

Test sequence	MED	Proposed algorithm		
		$N_{TH}=2$	$N_{TH}=4$	$N_{TH}=8$
Coastguard	62.2	48.9	45.5	43.6
Container	58.6	28.4	28.2	24.5
Mobile & Cal.	59.6	49.5	46.6	43.5
Flower	68.9	65.9	63.8	61.9
Mother	69.1	64.3	62.0	58.3
News	55.8	37.4	35.7	31.8
Foreman	67.6	74.5	72.5	69.2
Average	63.1	52.7	50.6	47.5

Conclusion: To reduce computational complexity, we have proposed a fast motion estimation algorithm. Identifying the boundary macroblocks associated with identical motion vectors, the proposed algorithm skips the motion vector search for inside macroblocks. Experimental results show that the proposed algorithm considerably reduces computational complexity at the cost of negligible PSNR degradation.

8 February 2006
Electronics Letters online no: 20060155
doi: 10.1049/el:20060155

S.-I. Park and I.-C. Park (*Division of Electrical Engineering, Department of Electrical Engineering and Computer Science, Korea Advanced Institute of Science and Technology, 373-1, Guseong-dong, Yuseong-gu, Daejeon 305-701, Republic of Korea*)

E-mail: icpark@ee.kaist.ac.kr

References

1 Dufaux, F., and Moscheni, F.: 'Motion estimation techniques for digital TV: a review and a new contribution', *Proc. IEEE*, 1995, **83**, (6), pp. 858–876
2 Yang, S., Wolf, W., and Vijaykrishnan, N.: 'Power and performance analysis of motion estimation based on hardware and software realizations', *IEEE Trans. Comput.*, 2005, **54**, (6), pp. 714–726
3 Chalidabhongse, J., and Kuo, C.-C.J.: 'Fast motion vector estimation using multiresolution-spatio-temporal correlations', *IEEE Trans. Circuits Syst. Video Technol.*, 1997, **7**, (3), pp. 477–488
4 Kang, H.-S.: 'Motion estimation algorithm with low complexity for video compression', *Electron. Lett.*, 2000, **36**, (18), pp. 1533–1535