

Modifications and Performance Improvements of 3-Step Search Block-Matching Algorithm for Video Coding

*Her-Ming Jong, Liang-Gee Chen, and Tzi-Dar Chiueh
Department of Electrical Engineering
National Taiwan University, Taipei, China.*

Abstract

This paper proposes three modifications to the 3-step hierarchical search block-matching algorithm for video coding: a multiple-winner search that improves the estimation accuracy, a method of subsampling that reduces computation and input data amount, and an overlapping strategy that improves the accuracy of large-area search. Experimental results show that combining these techniques provides high-speed and high-precision motion estimators with reduced on-chip buffers and lowered input bandwidth requirements.

1 Introduction

The block-matching motion estimation/compensation is widely used in several video coding standards. In motion compensated video coding systems, the present frame in the video sequence is divided into quadratic blocks. The process of block-matching is to find a candidate block, within a search area in the previous frame, that is most similar to the currently concerned block in the present frame. Full search (FS) block-matching exhaustively evaluates all possible displacements and provides the optimal solution. Its massive computation motivates developments of many fast block-matching algorithms (BMA's)[1]-[5]. These fast algorithms mainly use heuristic methods to decrease the number of locations for checking, or use subsampling to reduce the computation required for evaluating each candidate location. These algorithms significantly reduce the computation cost of block-matching process, and some of them provide sub-optimal accuracies close to that of full search.

Among these BMA's, the 3-step hierarchical search (3SHS)[1] performs well and is recommended by R-M8 of H.261[6] and SM3 of MPEG[7]. The purpose of this paper is to present three methods for improving accuracy and reducing input bandwidth and computation cost of this algorithm: multiple-winner hi-

erarchical search, search-area subsampling, and overlapped search for cases of large search area. These techniques can also be applied to other BMA's such as 2-D LOG[2], CS[3], etc., which use coarse-to-fine hierarchical searching schemes similar to that of 3SHS. These techniques are then combined to provide high precision and large search area motion estimators at costs much lower than that of full search.

2 Multiple-winner hierarchical search

Several hierarchical BMA's that follow a coarse-to-fine approach have been proposed[1]-[4]. In each step of these algorithms, some candidate locations around the winner of the previous step are checked. The one with the least distortion is then chosen as the center location of the next-step check points. This approach is developed based on the assumption that the distortion increases monotonically as the searched point moves away from the location of minimum distortion. However, this assumption is not always true for real-world sequences, and thus it makes the inappropriate choice in early steps excludes the optimal solution. A method for overcoming this drawback is to retain more than one winner for the next step. For example, two or more low-distortion candidate locations can be picked out as center locations of the next-step search. The trade-off between cost and performance is determined by experimental results presented in the following sections, which show that keeping two least-distortion locations in each step is appropriate for general video sequences.

3 Search-area subsampling

Subsampling is a well-known technique that reduces the computation in evaluating the distortion of a candidate block. However, direct subsampling (pixel decimation in both current blocks and search area) is prone to lose spatial resolution. The alternative subsampling technique[5] was thus proposed to improve the accuracy. This method preserves spatial details by using all pixels of current blocks and search area for calculating motion vectors. However, from the viewpoint of hardware implementation, this method saves computation but the requirements for I/O bandwidth and on-chip buffer remain unchanged.

¹This work is supported by National Science Council, Republic of China, under Grant NSC 82-0404-E-002-198.

Our observation suggests that, for general video sequences, the high interframe correlation makes details in two successive frames can be covered by preserving all pixels of only one frame (and the other frame is permitted to be appropriately subsampled). Because the search area is usually much larger than a current block, subsampling search area and keeping all current-block pixels can significantly reduce I/O and buffer requirements and preserve the accuracy. Fig.1 illustrates the subsampling strategy, in which different sampling masks are applied to current block when evaluating adjacent candidate locations.

4 Overlapped search for large search range

As proposed by MPEG SM3, the 3SHS can be directly expanded to more than three steps to cover a larger search range. But when the step number increases, the distances between checked points at early steps enlarge exponentially. Although the computation cost is much lower than FS of the same search range, the probability of being trapped in local minima significantly raises and the accuracy is thus dramatically reduced. To overcome this drawback, we proposed to use several independent 3SHS's, which inherently have search ranges of $-7 \sim +7$ pixels, to cover required larger search ranges. For example, the search range of $-15 \sim +14$ pixels can be covered by four 3SHS's. A weakness of this approach is that small motions cannot be detected well. This is because the smaller motion vectors locate near boundaries of the four composing 3SHS's, but 3SHS itself performs better for locations near center of its search range. The proposed solution is to apply an additional 3SHS that covers a search range centered at $(0, 0)$, as shown in Fig. 2. Although its search range overlaps that of other 3SHS's and results in some overhead, it significantly improves performance because small motion vectors frequently occur in typical video sequences.

5 Simulation Results

To decide the number of kept winners for MWHS, the percentage improvement defined as

$$\text{Percentage Improvement (\%)} = \frac{\text{PSNR(MWHS)} - \text{PSNR(3SHS)}}{\text{PSNR(FS)} - \text{PSNR(3SHS)}} \times 100\%$$

is evaluated for different winner numbers. Fig. 3 shows the average results of three and five benchmark sequences for H.261 and MPEG respectively: "salesman"(sales), "Miss America"(miss), "Claire"(cl), "Susie", "windmill"(wm), "table tennis"(tt), "football"(fb), and "mobile and calendar"(mob). These results suggest that two is an appropriate winner number by considering the trade-off

between cost and improvement.

Table 1~3 present performances and complexities of proposed methods and their combinations, together with some traditional approaches for comparison. In Table 1, all sequences are at a frame rate of 30 frame/sec, and search ranges are -8 (for FS) or -7 (for other algorithms) to $+7$ pixels. "Sub." stands for 1:2 subsampling of search-area pixels, and the MWH-S keeps two winners at both step 1 and 2 as centers of the next-step search range. An illustration is given in Fig.4, it shows that the performance improvement provided by MWHS is more significant than the degradation results from subsampling.

Table 2 and 3 present results of large-range searches for sequences of lower frame rates. In Table 2, test frames are temporally skipped by a factor of two to magnify interframe displacement. For all algorithms, search ranges also double in both horizontal and vertical directions. The algorithm "4x3SHS" merely contains four 3SHS's that cover the whole search range, and the OHS16 is the proposed overlapped scheme that provides an additional 3SHS to deal with small vectors. The direct expansion to four steps is also evaluated and labelled by "4SHS". The subsampling and MWHS can also applied to each 3SHS in OHS16 to provide further improvement(OHS16_{S.M.}). Results in Table 3 are similar to that in Table 2, except that interframe interval and search range are both enlarged by a factor of four, and thus 16 3SHS's are required to cover the search range. These results in Table 2~3 and Fig. 5~6 show that the proposed OHS is robust and performs close to full search. Computational complexities (relative to that of FS) of these algorithms are calculated by counting the number of required addition/subtraction (as shown in Table 1 ~ 3), which is a useful criteria when dedicated hardware is applied. Simulation results show that the proposed methods provide good trade-off between performance and cost.

6 Conclusion

The multiple-winner hierarchical search raises the accuracy of conventional 3-step hierarchical search block-matching algorithm; The search-area subsampling reduces the computation and input data amount; The proposed overlapped search enlarges the practical search range of hierarchical search algorithms. The combinations of these three proposed methods provide high-speed motion estimators that perform close to the optimal full search, even when the motion in image sequences and the required search ranges are enlarged.

References

- [1] T. Koga, K. Inuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing," in *Proc. Nat. Telecommun. Conf.*, New Orleans, LA, Nov.29-Dec.3, 1981, pp. G5.3.1-5.3.5.

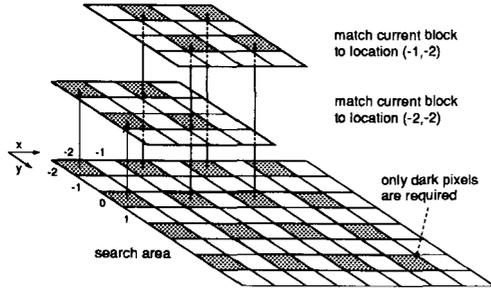


Figure 1: Subsampling the search area pixels.

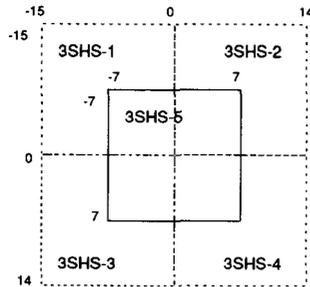


Figure 2: Search range overlapping of the proposed OHS16.

- [2] S. Kappagantula and K. R. Rao, "Motion compensated interframe image prediction," *IEEE Trans. Commun.*, vol. COM-33, no.9, pp.1011-1015, Sep. 1985.
- [3] M. Ghanbari, "The cross-search algorithm for motion estimation," *IEEE Trans. Commun.*, vol.38, no.7, pp.950-953, July 1990.
- [4] L. G. Chen, W. T. Chen, Y. S. Jehng, and T. D. Chiueh, "An efficient parallel motion estimation algorithm for digital image processing," *IEEE Trans. Circuits Syst. for Video Tech.*, vol.1, no.4, pp.378-385, Dec. 1991.
- [5] B. Liu and A. Zaccarin, "New fast algorithms for the estimation of block motion vectors," *IEEE Trans. Circuits Syst. Video Technol.* vol.3, no.2, pp.148-157, Apr. 1993.
- [6] CCITT SGXV, "Description of reference model 8 (RM8)," Document 525, Working Party XV/4, Specialists Group on Coding for Visual Telephony, June 1989.
- [7] MPEG, "ISO CD11172-2: Coding of moving pictures and associated audio for digital storage media at up to about 1.5Mbits/s", Nov.1991.

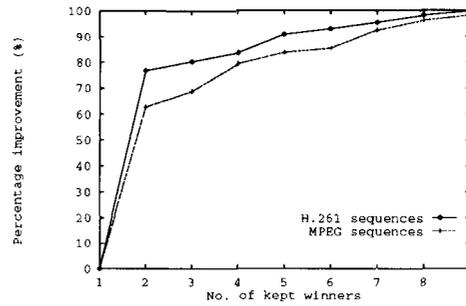


Figure 3: Percentage improvements of different winner numbers.

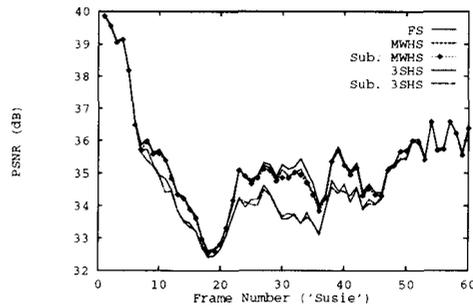


Figure 4: Comparison of various algorithms' PSNR's for normal frame rate and search range of $-8 \sim 7$ pixels.

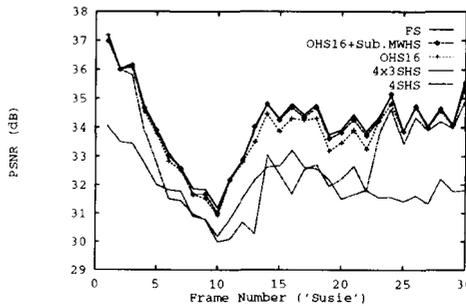


Figure 5: Comparison of various algorithms' PSNR's for 1/2 frame rate and search range of $-16 \sim 15$ pixels.

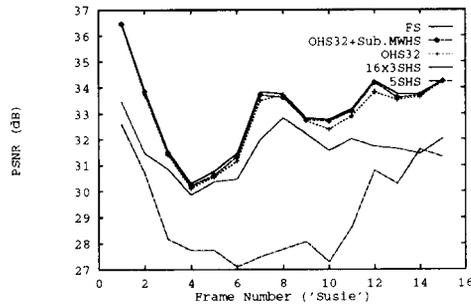


Figure 6: Comparison of various algorithms' PSNR's for 1/4 frame rate and search range of $-32 \sim 31$ pixels.

Algorithms	PSNR (dB)								Complexity (%)
	sales	miss	cl	susie	wm	tt	fb	mob	
FS	34.72	38.57	42.46	35.38	26.71	25.31	25.56	19.57	100
MWHS	34.68	38.42	42.45	35.34	26.65	24.58	25.36	19.51	18.2
Sub. MWHS	34.61	38.35	42.44	35.28	26.65	24.51	25.33	19.42	9.1
3SHS	34.53	38.19	42.29	34.90	26.41	24.04	25.27	19.40	11.1
Sub. 3SHS	34.46	38.10	42.28	34.85	26.40	23.95	25.24	19.30	5.6

Table 1: Simulation results of normal frame rate and search range of $-8 \sim 7$ pixels.

Algorithms	PSNR (dB)								Complexity (%)
	sales	miss	cl	susie	wm	tt	fb	mob	
FS16	34.91	38.10	39.69	34.11	24.57	24.60	26.00	19.39	100
OHS16 _{S.M.}	34.87	38.02	39.66	34.04	24.47	23.98	25.75	18.72	11.4
OHS16	34.86	38.01	39.66	33.87	24.29	23.92	25.66	18.38	13.9
4x3SHS	27.04	36.40	36.74	32.00	18.09	22.97	25.28	16.32	11.1
4SHS	34.55	37.32	39.15	32.85	22.03	22.38	25.45	17.76	3.7

Table 2: Simulation results of 1/2 frame rate and search range of $-16 \sim 15$ pixels.

Algorithms	PSNR (dB)								Complexity (%)
	sales	miss	cl	susie	wm	tt	fb	mob	
FS32	32.64	36.94	36.67	33.13	23.73	23.78	26.34	18.86	100
OHS32 _{S.M.}	32.61	36.85	36.64	33.04	22.28	23.28	26.20	18.32	9.7
OHS32	32.56	36.85	36.63	32.93	21.49	23.23	26.10	17.89	11.8
16x3SHS	26.79	36.17	35.10	31.60	18.62	22.85	25.72	16.60	11.1
5SHS	31.27	33.87	34.15	29.16	17.86	21.05	25.39	15.40	1.1

Table 3: Simulation results of 1/4 frame rate and search range of $-32 \sim 31$ pixels.