1998 International Computer Symposium
Workshop on Image Processing and Character Recognition
December 17-19, 1998, N.C.K.U., Tainan, Taiwan, R.O.C.

Hierarchical Motion Estimation Algorithm Based on Pyramidal Successive Elimination

Chia-Wen Lin
Computer and Communication Research Labs
Industrial Technology Research Institute
Hsinchu, Taiwan 310, R.O.C.
Email: ljw@n100.ccl.itri.org.tw

Yao-Jen Chang, and Yung-Chang Chen
Department of Electrical Engineering
National Tsing Hua University
Hsinchu, Taiwan 300, R.O.C.
Email: ycchen@ee.nthu.edu.tw

ABSTRACT

In this paper, we propose a fast motion estimation algorithm using pyramid hierarchy. Hierarchical Minkowski inequality is adopted to reduce the complexity of Mean Absolute Difference (MAD) computations. A top-town procedure is developed to search motion vectors hierarchically so that the total number of matching points is reduced drastically. The spatial/temporal correlation is also considered to further speed up the searching process. Experimental results show that the proposed algorithm can achieve high computation efficiency while maintaining comparable PSNR performance.

1. INTRODUCTION

Motion estimation plays a very important role in motion compensated coding algorithms. It's also the most time consuming operation in the codec system. The amount of computation required for MAD-based full search for motion estimation can take up to 70-80 % of the computing power of the whole encoding system [3]. Many research works on block-based motion estimation algorithms were conducted to reduce the computational cost in three ways: 1) fast search by reduction of the number of candidate blocks for matching [1-2]; 2) fast algorithm by reduction of the computational complexity of the matching criteria [3-6]; 3) fast algorithm by block motion field subsampling.

The mean absolute difference (MAD) is the most widely used matching criteria, the MAD of two $N \times N$ blocks X and Y is defined as

$$\mathrm{MAD}(X,Y)=\sum_{i=1}^{N}\sum_{j=1}^{N}\left|X(i,j)-Y(i,j)\right| \quad (1)$$

The Successive Elimination Algorithm (SEA) proposed in [1] adopted the well-known Minkowski inequality concept shown below

$$\left|(x_1+x_2)-(y_1+y_2)\right| \le \left|(x_1-y_1)\right| + \left|(x_2-y_2)\right| \quad (2)$$

to derive the following inequality:

$$\mathrm{MAD}(X,Y)\ge\left|\sum_{i=1}^{N}\sum_{j=1}^{N}X(i,j)-\sum_{i=1}^{N}\sum_{j=1}^{N}Y(i,j)\right| \quad (3)$$

Then, based on Equation (3), a fast search algorithm was developed in [1] which led to about three times faster than the Full-Search Algorithm (FSA). The Block Sum Pyramid Algorithm (BSPA) in [2] made extension of Equation (3) to a multiresolutional pyramid form. In the BSPA, the pyramid hierarchies for the candidate blocks of the previous frame and the template block of the current frame are firstly constructed. In each block sum pyramid hierarchy, as shown in Fig. 1, each pixel in the m-th level is the sum of 2 x 2 neighboring pixels in the $(m-1)$-th level, that is

$$X^m(i,j) = X^{m-1}(2i-1,2j-1)+X^{m-1}(2i-1,2j)+X^{m-1}(2i,2j-1)+X^{m-1}(2i,2j) \quad (4)$$

For an $N \times N$ block ($N = 2^M$), The m-th level MAD is defined as:

$$\mathrm{MAD}^m(X,Y)=\sum_{i=1}^{2^{M-m}}\sum_{j=1}^{2^{M-m}}\left|X^m(i,j)-Y^m(i,j)\right| \quad (5)$$

Then we can obtain the following multiresolutional Minkowski inequality [2].

$$\mathrm{MAD}^0(X,Y)\ge\mathrm{MAD}^1(X,Y)\ge\cdots\ge\mathrm{MAD}^{n-1}(X,Y)\ge\mathrm{MAD}^n(X,Y) \quad (6)$$

and

$$\mathrm{MAD}(X,Y)=\mathrm{MAD}^0(X,Y) \quad (7)$$

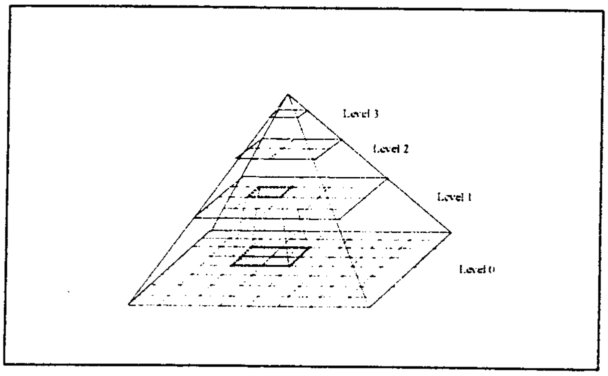

Fig. 1. Block Sum Pyramid Hierarchy for a 8 x 8 block

The pyramidal structure of the BSPA makes it more efficient in computation than the SEA, and both methods achieve the same performance with the full search algorithm (FSA). The BSPA is summarized as follows.

1. Select the motion vector of the corresponding block in the previous frame as the initial guess, and the MAD corresponding to the motion vector is chosen as the current MAD.
2. Construct the block sum pyramids for each candidate block in the search area of the previous frame.
3. Construct the block sum pyramid for the template block.
4. For a candidate block, compare its hierarchical MAD^m values in (5) with the current MAD and check the following.
   A. If the calculated MAD^m is larger than the current MAD, eliminate this candidate and got to Step 5.
   B. If the calculated MAD^m is less than the current MAD, set $m = m - 1$ and repeat Step 4 until down to the bottom level. Replace the current MAD with the calculated MAD at the lowest level ($m = 0$) and select this candidate block as the current match.
5. Repeat Step 4 for other candidate blocks until all the candidate blocks are compared.

2. THE PROPOSED FAST MOTION ESTIMATION ALGORITHMS

2.1 Hierarchical Motion Estimation Using Block Sum Pyramid

As mentioned above, the methods described in [1-2] just focused on reducing the cost of block-matching distortion computation. Though the computation cost can be effectively reduced using these two methods, they are still time consuming when the search area becomes large. In ITU-T H.263 standard the search area is a 32 x 32 window (normal mode) or a 64 x 64 window (unrestricted motion vector mode), which leads to up to 1024 and 4096 motion vector candidates respectively. The extra computation and memory costs to compute and store the block sum pyramids are also remarkable for large search area. Hierarchical search algorithms can drastically reduce the number of the searching candidates so as to reduce the aforementioned costs while maintaining comparable matching quality. In fact, the intrinsic multiresolution nature of the block sum pyramid algorithm makes it easy to be performed in a hierarchical manner.

In this paper, we propose an algorithm which takes advantage of both the high efficiency of matching criteria computation in BSPA and small number of matching candidates in hierarchical search algorithms. The proposed algorithm is described as follows.

1. Construct the block sum pyramids for each non-overlapping block in the search area of the previous frame.
2. Construct the block sum pyramid for the template block.
3. At the top level ($m = M$), search the block with minimum absolute difference, that is
   $$X_{I,J} = \arg\min_{i,j \in R^M} \mathrm{MAD}^M(T, X_{i,j})$$
   where R^M is the predefined search pattern at Level M. Then compute $\mathrm{MAD}^0(T, X_{I,J})$ and use it as the current MAD refernce: $\mathrm{MAD}_{current}$.
4. Search the best matching block with minimum MAD value within the reduced search grid R^m using the Steps 4-5 of BSPA described in Sec. 1.
5. *Set* $m = m-1$, shrink the search area and reduce the step size (2^m) then repeat Step 4. The size of the search area could be a function of the minimum MAD value obtained from the upper level. The simplest case is to halve both the horizontal and the vertical sizes used in the upper level.
6. Repeat Step 5 until it goes down to the bottom level.

2.2 Hierarchical Fast Search Based on the Spatial/Temporal Correlation

The study in [3] showed that there might often exist high spatial/temporal correlation for the motion vector values of adjacent blocks since they might belong to the same moving object and have similar motion behavior. Therefore it often makes sense to predict the motion vector value of the template block from the motion information of its spatially or temporally adjacent blocks. As shown in Fig. 2, we take into consideration the correlation between the motion of the template block and its spatially/temporally adjacent blocks to further speed up the searching process, and the Step 3 in the searching procedure mentioned in Sec. 2.1 is modified as follows.

3. Compute the MAD values with motion vectors corresponding to the adjacent blocks of the template block in the current frame and the previous frame. Then choose the motion vector with minimum MAD value as the initial guess and its associated MAD is set to be $\mathrm{MAD}_{current}$.
   A. If $\mathrm{MAD}_{current}$ is less than a predetermined threshold MAD_{th1}, the search process terminates and the motion vector obtained above is chosen as the best match.
   B. If $\mathrm{MAD}_{current}$ is larger than MAD_{th1}, but less than the other threshold MAD_{th2}, some existing fast algorithms well suited for small area search (e.g., BBGDS algorithm in [6])

could be combined with the block sum pyramid algorithm so as to form a very efficient search algorithm to search the best match.

C. If $MAD_{current}$ is larger than MAD_{th2}, the hierarchical procedure described in Steps 4-6 in Sec. 2.1 is used to search the best match.

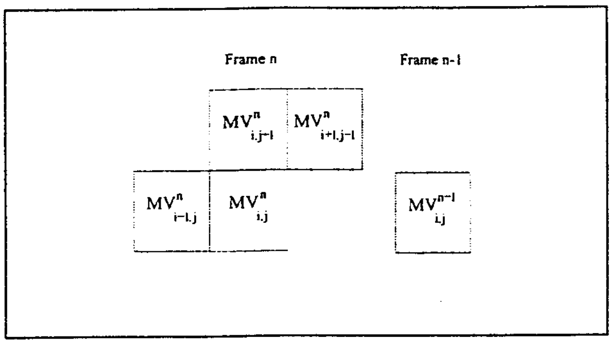

Fig. 2. The adjacent blocks with spatial/temporal correlation to the template block.

2.3 Complexity Analysis

The computational complexity analysis for the proposed algorithm is divided into two parts: 1) the cost for constructing the hierarchical block sum pyramids; 2) the cost for hierarchical matching. Assuming the image size is $W \times H$, it requires $\frac{3}{4m} W \times H$ to construct the block sum pyramids for all non-overlapping candidate blocks at the m-th level. With block size of 16 x 16, the number of levels is 4, the total number of addition operations is

$$\sum_{m=1}^{4} \frac{3}{4m} W \times H = \frac{75}{48} W \times H \quad (8)$$

which is much less than the number: $4(2W\text{-}1)(H\text{-}1)$ required for the BSPA [2]. For each template block, this computation overhead will cost only about 1.5 cadidate matching operations.

For $N \times N$ template block size, and $L \times L$ search area, The total number of matching cadidates required for the algorithm described in Sec. 2.1 depends on the search pattern defined for each level. If the search pattern used in the three step search (TSS) algorithm is adopted, only $\left(\frac{L}{N} - 1\right)^2 \times \log_2 N$ candidates will be matched for each template block which is much less than L^2 matching operarions performed in FSA, SEA, and BSPA. The number of search candidates can be further reduced by adopting the spatial/temporal correlation stated in Sec. 2.2. The memory cost required for the proposed algorithm to store the block sum pyramids is also much less than the BSPA since only a reduced set of block sum pyramids are computed and stored.

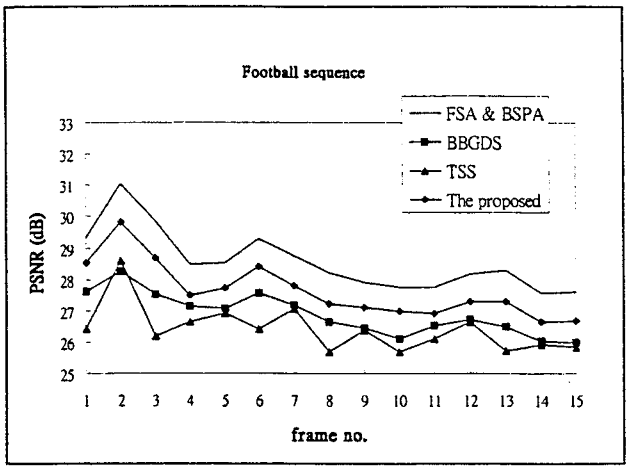

Fig. 3. Comparison of PSNR performance of various algorithms.

3. SIMULATION RESULTS

The PSNR performance comparison of our proposed algorithm with FSA, TSS, BBGDS, and BSPA is shown in Fig. 3. If the matching pattern used in TSS is adopted, the PSNR performance of our proposed algorithm in Sec. 2.1 is actually the same with TSS. Our method, however, achieve 2 to 5 times faster speed than the TSS depending on the motion statistics of the test image sequences. The modified algorithm proposed in Sec. 2.2 can achieve comparable PSNR performance for both slow and fast motion conditions compared to other methods, and the computing power required is pretty low. Table 1 shows the average PSNR performance for the three sequences: Salesman, Miss, and Football.

TABLE I Performance evaluation of various algorithms

	Salesman		Miss		Football	
Algorithm	DFD	PSNR	DFD	PSNR	DFD	PSNR
FSA	2.70	35.86	1.78	39.75	6.19	28.57
TSS	2.89	35.09	1.93	39.24	7.55	26.41
BSPA	2.70	35.86	1.78	39.75	6.19	28.57
BBGDS	2.75	35.63	1.94	39.33	7.00	26.89
The proposed	2.79	35.49	1.94	39.31	6.57	27.64

4. CONCLUSIONS

In this paper, we propose a fast motion estimation algorithm using hierarchical search based on pyramid hierarchy. Hierarchical Minkowski inequality is applied to reduce the complexity of Mean Absolute Difference (MAD) computations. A top-town procedure is developed to hierarchically search motion vectors from coarse to fine so that the total number of matching points is reduced drastically, which also reduces the computation and the memory costs for the construction of block sum pyramids. The spatial and temporal correlation is also taken into consideration to further speed up the searching process.

The experimental results show that the proposed algorithm can not only achieve higher computation efficiency but also provide comparable PSNR quality. The complexity analysis is also made to verify the computing power required for the proposed algorithm.

5. REFERENCES

[1] W. Li and E. Salari, "Successive Elimination Algorithm for Motion Estimation," *IEEE Trans. Image Processing,* vol. 4, pp. 105-107, Jan. 1995.

[2] C.-H. Lee and L.-H. Chen, "A Fast Motion Estimation Algorithm Based on the Block Sum Pyramid," *IEEE Trans. Image Processing,* vol. 6, pp. 1587-1591, Nov. 1997.

[3] B. Zeng, R. Li, and M. L. Liou, "Optimization of fast block motion estimation algorithms," *IEEE Trans. Circuits Syst. Video Technol.,* vol. 7, pp. 833-844, Dec. 1997.

[4] T. Koga, *et. al.,* "Motion-compensated interframe coding for video conferencing," in *Proc. Nat. Telecommunications Conf., New Orleans,* LA, Dec. 1981. pp. G5.3.1-G5.3.5

[5] J. R. Jain and A. K. Jain, "Diaplacement measurement and its application in interframe image coding," *IEEE Trans. Commun.,* vol. COM-29, pp. 1799-1808, Dec. 1981.

[6] L. -K. Liu and E. Feig, "A block-based gradient decent search algorithm for Block motion estimation in video coding," *IEEE Trans. Circuits Syst. Video Technol.,* vol. 6, pp. 833-844, Aug. 1996.