Published on International Computer Symposium 1998, October 1998

# Hierarchical Motion Estimation Algorithm Based on Pyramidal Successive Elimination

*Chia-Wen Lin*
Computer and Communication Research Labs
Industrial Technology Research Institute
Hsinchu, Taiwan 310, ROC

*Yao-Jen Chang, and Yung-Chang Chen*
Department of Electrical Engineering
National Tsing Hua University
Hsinchu, Taiwan 300, ROC

*ABSTRACT*

In this paper, we propose a hierarchical three step search algorithm using pyramid hierarchy. Hierarchical Minkowski inequality is adopted to reduce the cost of the sum of absolute difference (SAD) computations. A top-town procedure is developed to search motion vectors hierarchically and the search pattern used in the well-known Three Step Search (TSS) is adopted. We also propose a hybrid motion estimation scheme by combining the proposed Hierarchical Block Sum Pyramid Three Step Search (HBSPTSS) with the famous Block-Based Gradient Search (BBGDS) algorithms. Experimental results show that, compared to other existing search algorithms, the proposed hybrid search algorithm can achieve high computation efficiency for both slow-motion and fast-motion video contents while maintaining excellent PSNR performance. It is thus suitable for wide application fields with real-time requirement.

## 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 SAD-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 sum of absolute difference (SAD) is the most widely used matching criteria, the SAD of two $N \times N$ blocks $X$ and $Y$ is defined as

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

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

$$\left|(x_1 + x_2) - (y_1 + y_2)\right| \le \left|(x_1 - y_1)\right| + \left|(x_2 - y_2)\right| \tag{2}$$

to derive the following inequality:

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

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

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

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

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

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

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

and

$$\mathrm{SAD}(X,Y) = \mathrm{SAD}^0(X,Y) \tag{7}$$

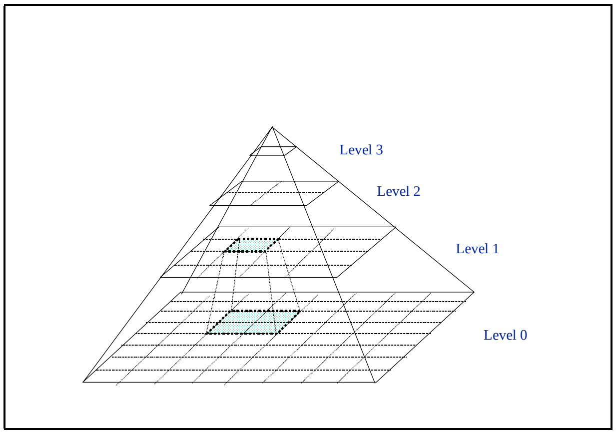



**Fig. 1**. Block Sum Pyramid Hierarchy for a 8 x 8 block

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 SAD corresponding to the motion vector is chosen as the current SAD.
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 $SAD^m$ values in (5) with the current SAD and check the following.
   A. If the calculated $SAD^m$ is larger than the current SAD, eliminate this candidate and got to Step 5.
   B. If the calculated $SAD^m$ is less than the current SAD, set $m = m - 1$ and repeat Step 4 until down to the bottom level. Replace the current SAD with the calculated SAD 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

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.

An image may include fast-motion and slow-motion objects simultaneously. Recently, some search algorithms were proposed for fast motion estimation based on the assumption that most of the objects' motions are zero or relatively small [6-7]. These algorithms have proven to have satisfactory performance on the slow-motion videos such as head-and-shoulders video in videophone applications. These fast algorithms, however, cannot treat the objects with fast motions very efficiently because the real motion vectors may be far away from the position of the target block. Adopting the motion vectors of the spatially/temporally neighboring blocks to predict the target block's motion vector can partly solve the problem since there may exist high spatial/temporal correlation among these motion vectors. There are, however, some exceptional cases where the motion vectors cannot be predicted well by simply adopting the spatial/temporal correlation followed by fast search algorithms suitable for small search range as in [6-7]. For example, in some video with many high-activity objects such as ball game, the spatial/temporal correlation among the motion vectors in some areas may be pretty low. A new fast algorithm which can adapt to various image contents is thus required.

### 2.1 Hierarchical Motion Estimation Using Block Sum Pyramid

Firstly, 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 Hierarchical Block Sum Pyramid (HBSP) algorithm is described as follows.

***The proposed HBSP algorithm:***

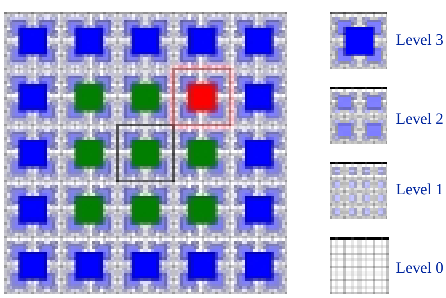
1. *Construct the block sum pyramids for each non-overlapping block in the search area of the previous frame.*
2. *Construct the block sum pyramid for the template block.*
3. *At the top level ($m = M$), search the block with minimum absolute difference, that is*

$$X_{I,J} = \arg \min_{i,j \in R^M} \mathrm{SAD}^M (T, X_{i,j})$$

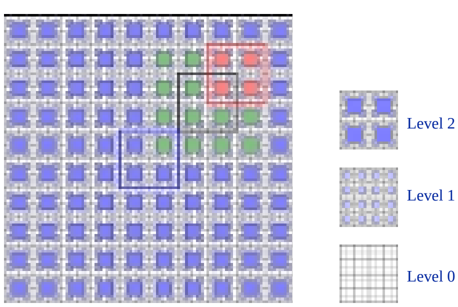
*where $R^M$ is the predefined search pattern at Level M. Then compute $SAD^0(T,X_{I,J})$ and use it as the current SAD refernce: $SAD_{current}$.*

4. *Search the best matching block with minimum SAD 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 SAD 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.*

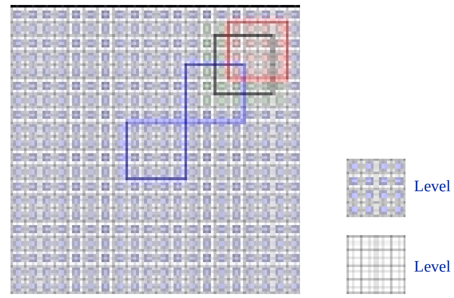
If the Three Step Search pattern is adopted for $R^m$, a Hierarchical Block Sum Pyramid Three Step Search (HBSPTSS) algorithm is formed. The detailed search steps are illustrated in Fig. 2. For simplicity and without loss of generality, the block size is assumed to be 8x8 in Fig. 2.



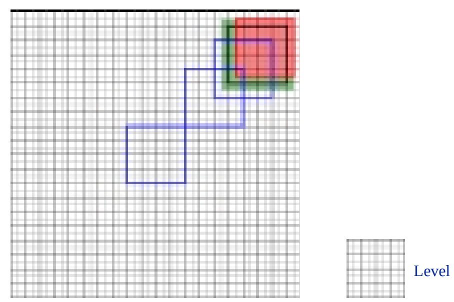

(a)




(b)




(c)




(d)

**Fig. 2**. The top-down hierarchical search steps of the proposed HBSPTSS for 8x8 block size (a) Level 3 (coarsest), (b) Level2, (c) Level 1, (d) Level 0 (finest).

### 2.2 Hierarchical Fast Search Based on the Spatial/Temporal Correlation

The study in [3] showed that there often exists 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. 3, we take into consideration the correlation between the motion of the template block and its spatially/temporally adjacent blocks to predict a proper initial search point to speed up the searching process. Most of the macroblocks' motion vectors can be predicted very well using the spatial/temporal correlation among the macroblocks, and some existing fast search algorithms can be adopted to estimate the motion vectors in small search areas very efficiently. (e.g., Block-Based Gradient Decent

Search (BBGDS) in [6]) However, in case a macroblock belongs to an different fast moving object from its neighboring macrobloks', the prediction method will fail to find a proper initial guess. In this case, the aforementioned HBPSTSS will be much efficient to fast estimate the motion vectors within a large search region. In this paper, we propose a hybrid scheme versatile to various image contents.

***<u>The proposed hybrid motion estimation scheme:</u>***

*Compute the SAD values with motion vectors corresponding to the neighboring blocks of the template block in the current frame and the previous frame. Then choose the motion vector with minimum SAD value as the initial guess and its associated SAD is set to be* $SAD_{current}$.

A. *If* $SAD_{current}$ *is less than a predetermined first threshold* $SAD_{th1}$, *the search process terminates and the motion vector obtained above is chosen as the best match.*
B. *If* $SAD_{current}$ *is larger than* $SAD_{th1}$, *but less than a second threshold* $SAD_{th2}$, *use BBGDS to search the best match.*
C. *If* $SAD_{current}$ *is larger than* $SAD_{th2}$, *use HBPSTSS to search the best match.*

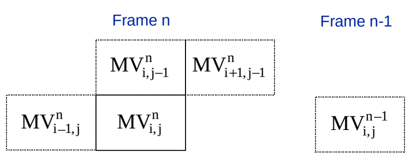



**Fig. 3**. The adjacent blocks with spatial and temporal correlation to the template block.

## 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 $\frac{3}{4^m} W \times H$ 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 per macroblock is

$$\frac{256}{W \times H} \sum_{m=1}^{4} \frac{3}{4^m} W \times H = 255 \tag{8}$$

which is much less than the number: $\frac{1024}{W \times H}(2W\text{-}1)(H\text{-}1) \cong 2048$ required for the BSPA [2]. For each template block, this computation overhead will cost only about 0.5 cadidate matching operations.

The total number of matching cadidates required for the proposed HBSPTSS is the same with the TSS algorithm which is much less than the matching operarions performed in FSA, SEA, and BSPA. The computation cost required for the HBSPTSS depends on the probabilities of in which levels the matching computations being finished. The probability distribution is highly related to the motion statistics of the image contents. A comparison of the computation costs between TSS and HBSPTSS is shown in Table. I. It's evident that, when compared to the TSS method, the HBSPTSS algorithm can achieve significane computation saving without sacrificing the performance. 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.

**TABLE I** Comparison of the computation costs of TSS and the proposed HBSPTSS

| | Miss_am | Salesman | Football |
|---|---|---|---|
| TSS | 100% | 100% | 100% |
| HBSPTSS | 83.7% | 45.3% | 67.8% |

The computation cost of the proposed hybrid motion estimation scheme can also be easily evaluated by computing the probability of each search strategy (no search, BBGDS, and HBSPTSS) and its associated computations (number of addtions/substractions).

## 3. SIMULATION RESULTS

The PSNR performance comparison of our proposed algorithm with FSA, TSS, BBGDS, and BSPA is shown in Fig. 4. If the proposed HBSPTSS is adopted, the PSNR performance is actually the same with TSS, while the HBSTSS achieves significant computation saving than the TSS depending on the motion statistics of the test image sequences as shown in Table. I. The proposed hybrid search algorithm with spatial/temporal correlation can achieve comparable PSNR performance for both slow and fast motion conditions compared to other existing methods, and the computing power required is pretty low. Table. II shows the average PSNR performance comparison of the proposed hybrid search scheme with FSA, BSPA, TSS, and BBGDS for the three sequences: Miss_am (slow motion), salesman (medium motion) and Football (high motion).

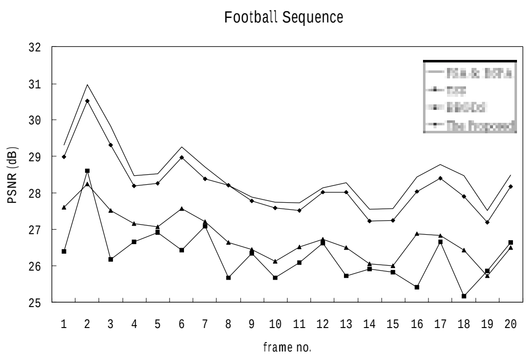



**Fig. 4**. Comparison of PSNR performance of various algorithms.

**TABLE II** Performance evaluation of various algorithms

| | Miss_am | | | Salesman | | | Football | | |
|---|---|---|---|---|---|---|---|---|---|
| Method | DFD | PSNR | Comp | DFD | PSNR | Comp | DFD | PSNR | Comp |
| FSA | 1.80 | 39.61 | 100 | 2.71 | 35.85 | 100 | 6.18 | 28.49 | 100 |
| BSPA | 1.80 | 39.61 | 56.14 | 2.71 | 35.85 | 21.26 | 6.18 | 28.49 | 23.52 |
| TSS | 1.95 | 39.07 | 3.34 | 2.88 | 35.11 | 3.34 | 7.59 | 26.30 | 3.31 |
| BBGDS | 1.95 | 39.22 | 1.27 | 2.76 | 35.63 | 1.11 | 7.02 | 26.79 | 1.63 |
| Proposed | 1.95 | 39.18 | 1.53 | 2.76 | 35.71 | 0.70 | 6.33 | 28.19 | 1.37 |

## 4. CONCLUSIONS

In this paper, we firstly propose a hierarchical three step search based on pyramid hierarchy. Hierarchical Minkowski inequality is applied to reduce the complexity of the sum of absolute difference (SAD) 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. To further reduce the computation cost, by combining with the proposed HBSPTSS method and the BBGDS method in [6] as well as exploiting the spatial/temporal correlation, we also propose a novel hybrid motion estimation scheme which can achieve high PSNR performance for both low and high motion video at extremely low computation cost. The proposed hybrid motion estimation scheme can thus be adopted in wide application fields with real-time requirement.

## 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.

[7] 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, pp. 3133-316, Jun. 1996.