

# A New Fast Motion Estimation Algorithm using Hexagonal Subsampling Pattern and Multiple Candidates Search

*K.T. Choi, S.C. Chan and T.S. Ng*

Department of Electrical and Electronic Engineering

The University of Hong Kong

Email: scchan@hkueee.hku.hk

## Abstract

In this paper we present a fast algorithm to reduce the computational complexity of block motion estimation. The reduction is obtained from the use of a new hexagonal subsampling pattern and the domain decimation method introduced in [14]. The multiple candidates search method is also introduced to improve the robustness of the algorithm. Computer simulation shows that the performance is very close to that of the full search.

## I. Introduction

Motion estimation/compensation is an effective method for reducing temporal redundancy found in video sequence compression. Due to its simplicity, block matching algorithm (BMA) is frequently used to estimate the motion of a given block to be encoded. In BMA, all pixels in a block is assumed to undergo a translational motion. An estimate of the current block can therefore be obtained by searching similar blocks in the previous encoded (or original image) frame around the current block. In most video coding standards like H.261 and MPEG, the block size for motion estimation is (16x16). A straight forward but computational very expensive mean is to use the full search algorithm (FS) which estimates motion vectors by searching all the points in the search area. A candidate block in the search frame is shifted along all the positions within the search area to find the best match to the reference block of the current frame. There are a number of matching criterion for BMA but the mean absolute difference (MAD) is most commonly used because no squaring operation is required. The MAD measure is defined as

$$MAD(x,y) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} |f(x,y) - f'(x+i,y+j)| \quad (1)$$

where  $M \times N$  is the block size,  $f(x,y)$  is the reference block of the current frame and  $f'(x+i,y+j)$  is the candidate block within the search area. Motion vector is the value  $(i,j)$  which results in the minimum value of  $MAD(i,j)$  within the search area. It is optimal in terms of the performance measure specified, but its computational

complexity is very high. For a 352x240 picture with a block size of 16 x 16 and a search range of  $\pm 15$  pixels, the complexity of the full search can be as high as  $(16 \times 16) \times (31)^2 \times 330 \approx 81 \times 10^6$  operations per frame. Therefore, in order to reduce the complexity of the encoding system, it is necessary to develop more efficient motion estimation algorithms.

Several fast BMAs have been developed in the past to alleviate the heavy computation of FS. These include three-step hierarchical search (3SHS) [1][6], 2-D logarithmic search (LOGS) [4], one-at-a-time search (OTS) [7], etc. These algorithms reduce the number of computations by either reducing the number of locations searched or the number of arithmetic operations in each comparison. However, as the MAD surface often has several minima, this might make an inappropriate choice in early steps and subsequently exclude the global optimal motion vector. In pixel decimation algorithm [2][3], the current block is subsampled by a factor of two in both the horizontal and vertical direction to give a four-to-one subsampling ratio during the motion search. To reduce the aliasing due to the subsampling, the four possible subsampling patterns in each block are used alternately over the search area. The arithmetic complexity is therefore reduced by a factor of four.

In this paper, a new fast block matching algorithm is presented. This algorithm makes use of a new Pixel Decimation Pattern and the domain decimation algorithm [14] to achieve a computation reduction factor of 16 while having close-to-optimal performance in the PSNR and MAD. The domain decimation try to reduce the arithmetic complexity by reducing the number of locations searched and the arithmetic operations in each comparison. To increase the robustness of the algorithm, we also make use of a technique called multiple candidates search introduced in [14]. The paper is organized as follows. In Section 2 the new sampling pattern and the domain decimation method are introduced. Section 3 presents simulation results for the proposed algorithm. Finally a conclusion is given in Section 4.

## II. Proposed Algorithm

Subsampling of pixel block to reduce computational complexity of BMA has been previously proposed in the literature. Bierling [6] introduced a hierarchical motion-estimation technique in which a fast estimation of motion field is found from low-pass filtered and subsampled images. Previously, Koga[1] et al. subsampled unfiltered image by a factor of 2. Also, in the pixel-decimation algorithm Liu[2][3] proposed to use unfiltered images. In this paper, we use the unfiltered image with a different subsampling pattern on the domain block.

### A) Hexagonal Subsampling Pattern

In the pixel decimation method[2,3], the current block are decimated to reduce the arithmetic complexity of each comparison. Figure 1 shows a block of 8x8 pixels for illustration. Pattern A is the subsampling pattern that consists of all a pixels. Similarly, patterns B, C, and D are subsampling patterns that consist of all the b, c, and d pixels, respectively. Since only the pixels of pattern A are used for matching, the computation is reduced by a factor of 4. To reduce the drawback of pixel subsampling, all four subsampling pattern are then alternatively used over the search area. If pattern A is used at the location  $(x,y)$ , then within the area, it is also used at location  $(x+2i,y+2i)$  for  $i,j$  integers. Patterns B, C, and D will be used at locations  $(x+1+2i,y+2i)$ ,  $(x+1+2i,y+1+2i)$ , and  $(x+2i,y+1+2i)$ , respectively. In hexagonal subsampling pattern, the pattern A is chosen from the hexagonal sampling lattice as shown in figure 1(b). Hexagonal sampling is similar to the rectangular sampling except that it is shifted by one pixel in alternate row of the domain block. We have simulated the original and proposed pattern in decimation search using 10 frames of the 'Table Tennis' sequence (352 x 240 pixels) with error signal coding. The maximum displacement in the search space was  $\pm 16$  pixels in both the vertical and horizontal directions. The frame pattern is IPPPP, where I represents the intra mode frame (or I frame) and P represents the predict mode frame (or P frame) as defined in MPEG-1 standard. The average Peak Signal to Noise Ratio (PSNR) and bit per pixel (bpp) after encoding for each search method are noted. Experimental results show that the proposed hexagonal pattern performs slightly better than rectangular pattern (see table 1).

### B) Domain Decimation

In pixel decimation method [2,3], the current block is decimated to reduce the arithmetic complexity of each comparison. However, the number of location searched is

still the same, i.e.  $(2p+1)^2$  for a search range of  $\pm p$  pixels. If both the current block and the encoded (or original) picture are decimated by a factor of two in the horizontal and vertical directions, then the number of candidates to be searched is only  $(2\lfloor p/2 \rfloor + 1)^2$  where  $\lfloor i \rfloor$  denotes the integer just less than  $i$ . Due to the decimation of the search domain, the precision of motion vector is  $\pm 1$ . Therefore, the full search in the decimated domain can only provide an initial guess for the displacement and a full pel search in the original domain with a search range of at least  $\pm 1$  is needed. Greater search range instead of  $\pm 1$  can be used because the additional operations required is very low as compared with the remaining. This would reduce the arithmetic operations by a factor of approximately 16. To reduce the aliasing due to the subsampling, the four possible subsampling patterns in each block are used alternatively over the search area. The computational complexity is then reduced by another factor of 4.

### C) Multiple Candidates Search

Since the MAD surface can have several minima, the above algorithm may get trapped in local minima. Here, we employ the multiple candidates search in which we keep more than one winner for the next search. For each subsampling pattern, we keep the best  $n$  candidates for the next search. In total, we have kept  $4n$  candidates and a final reduced full search with a search window size of  $\pm 1$  pixel is performed for each candidate. The best candidate is then taken as the result. The incorporation of the multiple candidates search at the final search increase the robustness of our algorithm. The experimental result of the proposed algorithm is very close to the optimal as compared to the Full Search (FS).

## III. Experimental Results and Discussions

We have done 2 simulations here. In the first simulation, we implement our algorithm without coding. Our algorithm was simulated using 50 frames of sequence 'Football' with 352x240 pixels. Only the luminance component was considered. The maximum displacement in the search space was  $\pm 15$  pixels in both the vertical and horizontal directions with a block size of 16x16 and 4 candidates for each pattern. We compared the algorithms using the mean square prediction error (MSE) of the motion-compensated frames. As seen in Fig.2, the result of the proposed algorithm is comparable to the standard 4-to-1 pixel decimation but with lower complexity.

In the second simulation, we implemented the proposed algorithm in Berkeley MPEG-1 encoder. A total of 110 frames of 'Table Tennis' sequence with 352x240 pixels was tested. The frame pattern used is IPPP, where I represents the intra mode frame (or I frame) and P represents the predict mode frame (or P frame) as defined in MPEG-1 standard. The maximum displacement in the search space was  $\pm 15$  pixels in both the vertical and horizontal directions with a block size of 16x16. Figure 3 shows the mean square error comparison of the decoded pictures at a bitrate of 1 Mbps. It is observed that the performance of the proposed algorithm is very close to the coder using Liu's and the full search algorithm. In terms of objective and subjective image quality, the estimated frames of the proposed method is also very close to the exhaustive search and standard 4-to-1 pixel decimation algorithm.

#### IV. Conclusion

In this paper, we present a new fast motion estimation algorithm using Domain Decimation, Hexagonal Subsampling Pattern and Multiple Candidates Search. The computational complexity of this algorithm can be reduced by a factor of 16 when compared to that of FS. Computer simulations show that the algorithm is very efficient as compared with full search, with a significant reduction on computation. It also is fully compatible with H.261 and MPEG-1 standard, and can easily be combined with other hierarchical search strategy[14].

#### Acknowledgments

This work is supported by the Hong Kong Research Grants Council and the CRCG of the University of Hong Kong.

#### 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., pp. G5.3.1-5.3.5, New Orleans, LA, Nov. 29-Dec. 3 1981.  
 [2] B.Liu and A.Zaccarin, "New Fast Algorithms for Estimation of Block Motion Vectors," IEEE Trans. Circuits and Systems for Video Technology, vol. 3, No.2, April 1993, pp 148-157.  
 [3] A.Zaccarin and B.Liu, "Fast Algorithms for Block Motion Estimation", Proc. IEEE ICSPAAP-92, pp. III-449-452.  
 [4] J.R.Jain and A.K. Jain, "Displacement measurement and its application in interframe image coding," IEEE Trans. Commun., vol. COM-29, Decemember 1981, pp. 1799-1808.  
 [5] M.Ghanbari, "The cross-search algorithm for motion estimation," IEEE Trans, Comm., vol. COM-38, no. 7 pp. 950-953, July 1990.

[6] M.Bierling, "Displacement estimation by hierarchical block matching," in SPIE Visual Commun. Image Processing '88, vol. 1001, pp. 942-951, 1988.  
 [7] R.Srinivasan and K.R.Rao, "Predictive coding based on efficient motion estimation," ICC'84, pp521-526.  
 [8] S.Zfar and Y.Zang, and J.S.Baras, "Predictive block-matching estimation for TV coding - Part I: Interblock prediction," IEEE Trans. Broadcasting, vol.37, no. 3 pp. 97-101, Sept. 1991.  
 [9] Y.Zang and S.Zafar, "Predictive block-matching estimation for TV coding - Part II: Interblock correlation," IEEE Trans, Broadcasting, vol.37, no. 3, pp102-105, Sept. 1991.  
 [10] A.Puri, H.M.Hang, and D.L. Schilling, "An efficient block matching algorithm for motion compensated coding," in Proc. Int. Conf. Acoust., Speech, Signal Processing, 1987, pp. 25.4.1-25.4.4.  
 [11] J.S.Kim and R.H.Park, "A fast feature-based block matching algorithm using integral projection," IEEE Journal on Selected Areas in Communications, vol, 10, no. 5, pp.968-979, June 1992.  
 [12] H.M.Jong, L.G.Chen, and T.D.Cheun, "Accuracy improvement and cost reduction of 3-step search block matching algorithm for video coding", IEEE Trans. Circuits Syst. Video Technol., 4(1), pp.88-90, 1994.  
 [13] K.M.Nam, J.S.Kim, R.H.Park, and Y.S.Shim, "A fast hierarchical motion vector estimation algorithm using mean pyramid," IEEE Trans. Circuits Syst. Video Technol., vol.5, no.4, pp.343-351, August 1995.  
 [14] K.W.Cheng and S.C.Chan, "Fast Block Matching Algorithms for Motion Estimation," Proc. IEEE ICASSAP-96, Atlanta, vol 4, pp 2313.

Algorithm	Bits per Pixel	Average PSNR
Exhaustive	0.2352	30.5
Rectangular	0.2383	30.5
Hexagonal	0.2377	30.5

Table 1: Performance comparisons for exhaustive search, pixel decimation search using rectangular and hexagonal pattern (proposed).

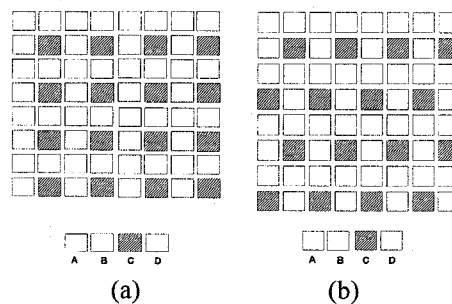


Fig. 1: (a) The rectangular sampling pattern[2][3] and (b) the proposed hexagonal sampling pattern in search domain.

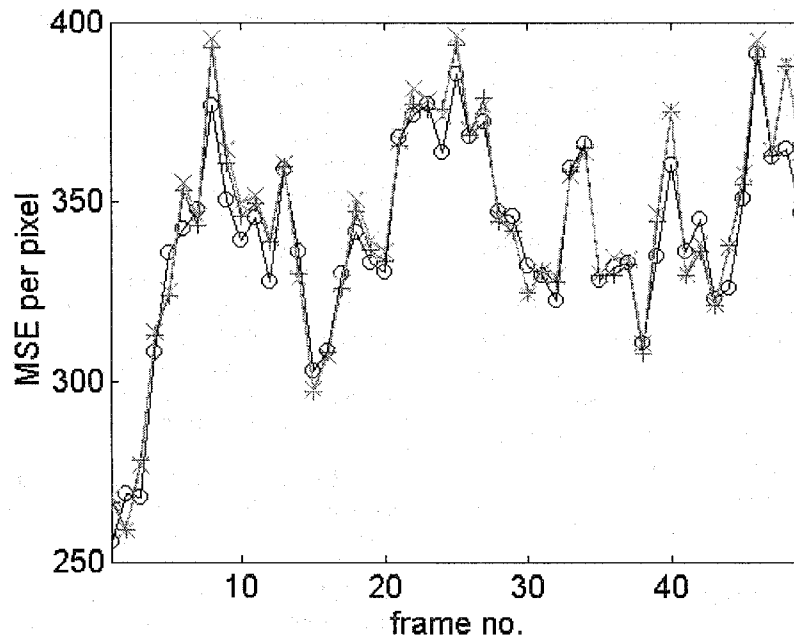


Figure 2: MSE per pixel for “Football” Sequence (352x240) using different algorithms for 50 frames. The o line for the Proposed method, + line for Full search, and x line for Standard Pixel Decimation. Simulation is performed with 16x16 block size without coding the error signal.

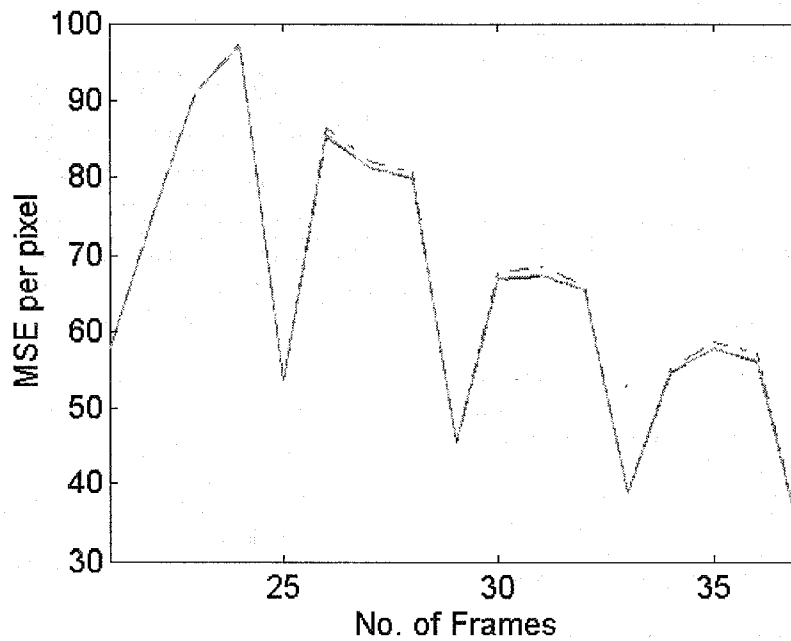


Figure 3: MSE per pixel for “Table Tennis” Sequence (352x240) using different algorithms from frame no. 23 to 38. Solid line for the Proposed method, dashed line for Full search method, and dashed-dot line Standard Pixel Decimation. Simulation is performed with 16x16 block size with error signal coding at 1 Mbps.