An Adaptive Block Matching Algorithm based on Temporal Correlations

Hyo-Sun Yoon, Nam-Rye Son, Guee-Sang Lee and Soo-Hyung Kim

Department of Computer Science, Chonnam National University,
300 Youngbong-dong, Buk-gu, Kwangju 500-757, Korea
Tel.:+82-62-530-0147, Fax.:+82-62-530-3439
Email:estheryoon@hotmail.com, {nrson,gslee,shkim}@chonnam.chonnam.ac.kr

Abstract: To reduce the bit-rate of video sequences by removing temporal redundancy, motion estimation techniques have been developed. However, the high computational complexity of the problem makes such techniques very difficult to be applied to high-resolution applications in a real time environment. For this reason, low computational complexity motion estimation algorithms are viable solutions. If *a priori* knowledge about the motion of the current block is available before the motion estimation, a better starting point for the search of an optimal motion vector can be selected and also the computational complexity will be reduced. In this paper, we present an adaptive block matching algorithm based on temporal correlations of consecutive image frames that defines the search pattern and the location of initial starting point adaptively to reduce computational complexity. Experiments show that, comparing with DS(Diamond Search) algorithm, the proposed algorithm is about $0.1 \sim 0.5$(dB) better than DS in terms of PSNR and improves as much as 50% in terms of the average number of search points per motion estimation.

1. Introduction

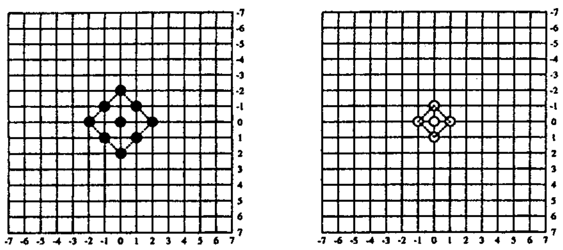
Recently, great interest has been devoted to the study of different approaches in video compressions. The high correlation between successive frames of a video sequence makes it possible to achieve high coding efficiency by removing the temporal redundancy. Motion estimation (ME) and motion compensation techniques are an important part of most video encoding, since it could significantly affect the compression ratio and the output quality. The most popular motion estimation and motion compensation method has been the block-based motion estimation, which uses a block matching algorithm (BMA) to find the best matched block from a reference frame. ME based on the block matching are adopted in many existing video coding standards such as H.261/H.263 and MPEG-1/2/4. If the performance in terms of prediction error is the only criterion for BMA, full search block matching algorithm(FS) is the simplest BMA, guaranteeing an exact result. FS can achieve optimal performance by examining all possible points in search area of the reference frame. However, FS is very computationally intensive and it can hardly be applied to any real time applications. Hence, it is inevitable to develop fast motion estimation algorithms for real time video coding applications. Many low complexity motion estimation algorithms such as Diamond Search (DS) [1], [2], Three Step Search(TSS)[3], New Three Step Search(NTSS)[4], Four Step Search(FSS)[5], Two Step Search(2SS)[6] and Two-dimensional logarithmic search algorithm[7] have been proposed. Regardless of the characteristic of the motion of a block, all these fast block matching algorithms(FBMAs) use fixed search patterns, which results in the use of many checking points to find a good motion vector(MV). Since this class of BMAs do not have any information on the motion of the current block, they use the origin of the search window as the starting point. To improve the accuracy of the existing BMA algorithms, in this paper, the motion correlation between successive frames is used to predict an initial starting point that reflects the current block's motion trend. Because a properly predicted initial starting point makes the global optimum closer to the predicted starting point, it increase the chance of finding the optimum or near-optimum motion vector with less search points.

In this paper, we proposed an adaptive block matching algorithm based on temporal correlations. In this algorithm, the motion vector of the block with the same coordinate in the reference frame is considered to predict the movement of the current block. And then we determine an initial starting point and the search pattern adaptively according to the predictive movement of the current block.

This paper is organized as follows. Section 2 describes the existing motion estimation algorithms. The proposed algorithm is described in Section 3. Section 4 reports the simulation results and conclusions are given in Section 5.

2. Motion Estimation algorithms

Among many strategies for motion estimation, in particular, we focus on the block-based scheme where the frames are sub-divided into smaller units called blocks. The block-based motion estimation uses BMA to find the best matched block from a reference frame. The displacement between the best-matched block in the reference block and the current block is called a motion vector(MV). Many search algorithms for motion estimation have been developed[1–9]. FS, the simplest algorithm, examines every point in the search area to find the best



(a) LDSP (b) SDSP

Figure 1. Diamond Search algorithm(DS)

match. Clearly, it is optimal in terms of finding the best motion vector, but it requires very high computations. This has led to the development of FBMAs such as DS, TSS, NTSS, FSS and 2SS. The TSS is a coarse-to-fine search algorithm. The starting step size for search is large and the center of the search is moved in the direction of the best match at the stage, and the step size is reduced by half. In contrast, FSS starts with a fine step size(usually 2) and the center of the search is moved in the direction of the best match without changing the step size, until the best match at that stage is the center itself. The step size is then halved to 1 to find the best match. In other words, in FSS the search process is performed mostly around the original search point (0,0), or it is more center-biased. Based on the characteristics of a center-biased motion vector distribution, NTSS enhanced TSS by using additional search points, which are around the search origin (0,0) of the first step of TSS. The DS is also a center-biased algorithm by exploiting the shape of the motion vector distribution. Regardless of the characteristic of the motion of a block, DS shows the best performance compared to these FBMA methods in terms of both average number of search points per motion vector and the PSNR (peak signal to noise ratio) of the predicted image. The DS algorithm is summarized as follows.

The DS algorithm uses two diamond search patterns, depicted in Figure 1. At first, the large diamond pattern (LDSP) is used for the gradient-based coarse search. When the centered search position of LDSP shows the minimum block distortion, the small diamond search pattern (SDSP) is chosen for fine search. And when the centered search position of SDSP shows the minimum block distortion, this algorithm terminates.

3. The proposed algorithm

If we can use any information on the motion of the current block before the motion estimation, the new starting point for the current can be set so that we can find the motion vector with much less number of search points. Since the time interval between successive frames is very short, there are high temporal correlations between successive frames of a video sequence.

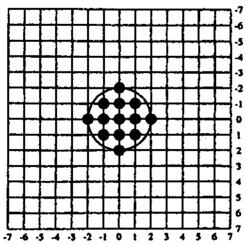
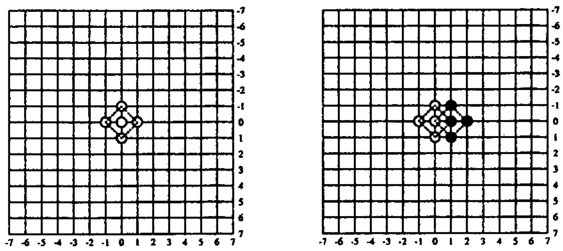


Figure 2. Motion vector distribution

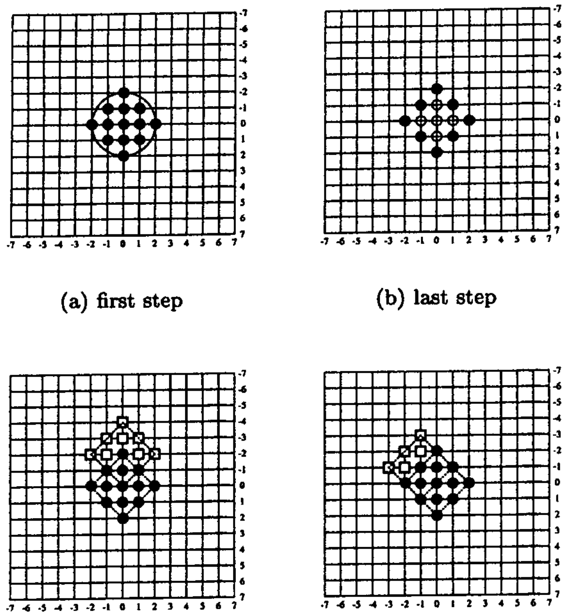


(a) first step (b) second step

Figure 3. Small Diamond Search Algorithm(SDSP)

In other words, the motion of the current block is very similar to that of the same coordinate block in the reference frame. In this paper, we use the MV of the same coordinate block in the reference frame to determine the starting point of the search and then the search patterns are selected adaptively.

In our algorithm, depending on the MV of the same coordinate block in the reference frame, two search patterns as illustrated in Figure 3 and Figure 4 are used. If the MV of the same coordinate block in the reference frame is zero vector as in a stationary block, the search pattern called small diamond search pattern (SDSP)[8] as shown in Figure 3 is used without changing the starting point which is the search origin (0,0). In Figure 3(a), white circles are the initial search points and in Figure 3(b), black circles are search points added in the second step. Note that the center of black circles is the position which showed the minimum distortion in the first step. Otherwise, the starting point is moved to the displacement of the MV of the same coordinate block in the reference frame and then the modified diamond search pattern (MDSP)[9] as illustrated in Figure 4. is used for motion estimation. Based on the fact that about 50%(in large motion case) $\sim$ 98%(in small motion case) of motion vectors are enclosed in a circular support, as shown in Figure 2, with a radius of 2 pixels[1], [2] and centered on the search origin (0,0), the circular support around the starting point becomes the initial search points in MDSP as shown in Figure 4(a). If one of the positions notated with the '$\oplus$'s in Figure 4(b) shows the minimum distortion among the search points of the first step of Figure 4(a), the search procedure terminates. Otherwise, the new search points are



(a) first step (b) last step

(c) corner point (d) edge point

Figure 4. Modify Diamond Search Algorithm(MDSP)

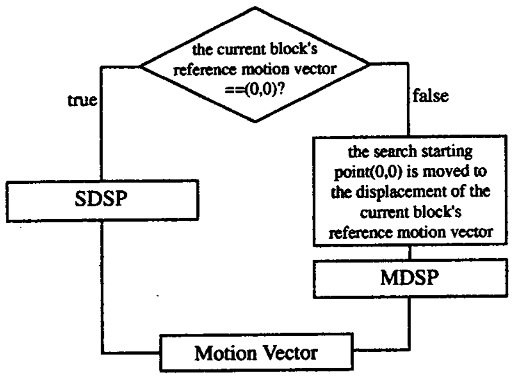


Figure 5. The block diagram of the proposed algorithm

set as shown in Figure 4(c) or Figure 4(d).

The block diagram of the proposed algorithm appears in Figure 5. According to the current block's reference motion vector, the proposed algorithm selects adaptively the search pattern between SDSP and MDSP. If the current block's reference MV is zero vector, SDSP is selected for the motion estimation. Otherwise, MDSP is chosen. The proposed algorithm is summarized as follows.

Step 1 If the current block's reference motion vector is zero vector, go to Step 2; otherwise, go to Step 3.

Step 2 When the current block's reference motion vector is zero vector, then

I. The initial SDSP is centered at the origin of search window, and the 5 checking points of SDSP as seen in Figure 3(a) are tested. If the minimum block distortion (MBD) point calculated is located at the center position of SDSP, then it is the final solution of the motion vector. otherwise go to II.

II. The MBD point founded in the previous search step is repositioned as the center point to form a new SDSP. If the new MSD point obtained is located at the center position, then it is the final solution of the motion vector. Otherwise, recursively repeated this step.

Step 3 When the current block's reference motion vector is non-zero vector, then

I. The starting point(0,0) is moved to the displacement of the current block's reference motion vector, let's call the moved search starting point the new starting point.

II. The MDSP is disposed at the center of the new search starting point, and the 13 checking points of MDSP as seen in Figure 4(a) are tested. If the MBD point calculated is located at the center position of MDSP or one of $\oplus$ points in Figure 4(b), then it is the final solution of the motion vector. otherwise go to III.

III. If the MBD point is located at the corner of MDSP, eight additional checking points as shown in Figure 4(c) are used. If the MBD point is located at the edge of MDSP, five additional checking points as shown in Figure 3(d) are used. And then the MBD point found in the previous search step is repositioned as the center to from a new MDSP. If the minimum block distortion(MBD) point calculated is located at the center position of MDSP or one of $\oplus$ points in Figure 4(b), then it is the final solution of the motion vector. Otherwise, recursively repeated this step.

4. Simulation result

In this section, we show the experiment results for the proposed algorithm. We compared FS, NTSS, FSS, 2SS and DS to the proposed algorithm in both of image quality and search speed. Eight QCIF test sequences are used for the experiment: Suzie, Foreman, Mother and Daughter, Carphone, Salesman, Stefan, Table, Claire. The mean square error(MSE) distortion function is used as the block distortion measure(BDM). The quality of the predicted image is measured by the peak signal to noise ratio(PSNR), which is defined by

$$MSE = \left(\frac{1}{MN}\right) \sum_{m=1}^{M} \sum_{n=1}^{N} [x(m,n) - \hat{x}(m,n)]^2 \quad (1)$$

$$PSNR = 10\, log_{10} \frac{255^2}{MSE} \quad (2)$$

In Eq. (1), $x(m,n)$ denotes the original image and $\hat{x}(m,n)$ denotes the motion compensated prediction image. From Table 1 and 2 we can see that proposed algorithm is better than DS in terms of both the computational complexity (as measured by the average number of search points per motion vector) and PSNR of the predicted image. In terms of PSNR, the proposed

Table 1. Average PSNR of the test image sequence

	FS	NTSS	FSS	2SS	DS	Proposed
Stefan	23.88	22.24	22.62	23.85	22.77	23.16
Foreman	29.54	28.19	28.22	29.24	28.66	29.06
Suzie	32.19	31.64	31.54	32.16	31.89	32.08
Table	26.50	25.60	24.81	26.27	25.67	25.70
Carphone	30.88	30.74	30.15	30.77	30.48	30.70
Salesman	32.70	32.69	32.53	32.70	32.62	32.69
Claire	35.05	34.91	34.74	35.01	34.85	34.93
M&D	31.52	31.47	31.34	31.51	31.42	31.49

Table 2. Average number of search points per motion vector estimation

	FS	NTSS	FSS	2SS	DS	Proposed
Stefan	961	20.06	18.94	255	16.24	8.95
Foreman	961	19.39	18.66	255	15.41	7.22
Suzie	961	18.65	17.84	255	14.41	7.00
Table	961	19.78	18.70	255	15.50	8.22
Carphone	961	18.62	17.83	255	14.44	6.89
Salesman	961	17.15	17.05	255	13.09	5.17
Claire	961	17.24	17.08	255	13.15	5.20
M&D	961	17.37	17.15	255	13.27	5.44

algorithm is about 0.1(dB) better than DS in stationary sequences such as Suzie, Salesman, Claire, Mother and Daughter and about 0.5(dB) in motioned sequences such as Stefan, Foreman, Table, Carphone in Table 1. In terms of the average number of search points per motion vector, the proposed algorithm improves as high as 50% compared with DS. The 2SS shows the performance in PSNR very close to our algorithm, but the proposed algorithm requires less computation by up to more than 95% on average as shown in Table 2.

5. Conclusion

Based on the temporal correlation between successive frames, an adaptive block motion algorithm is proposed in this paper. The proposed algorithm chooses the search pattern and the search starting point according to the motion vector of the same coordinate block in the reference frame. Through experiments, compared with DS, the proposed algorithm is about 0.1 $\sim$ 0.5 (db) better than DS in terms of PSNR and improves as high as 50% compared with DS in terms of average number of search points per motion vector. The proposed algorithm reduces the computational complexity compared with previously developed fast BMAs, while maintaining better quality.

6. Acknowledgment

This work was supported by Brain Korea 21 Project.

References

[1] J.Y. Tham, S. Ranganath and A. A. Kassim, "A Novel Unrestricted Center-Biased Diamond Search Algorithm for Block Motion Estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 8, No. 4, pp. 369-375, 1998.

[2] Z. Shan and M. Kai-kuang, "A New Diamond Search Algorithm for Fast block Matching Motion Estimation," *IEEE Transactions on Image Processing*, Vol. 9, No. 2, pp. 287-290, 2000.

[3] T. Koga, K. Iinuma, Y. Hirano, Y. Iijim and T. Ishiguro, "Motion compensated interframe coding for video conference," *In Proc. NTC81*, pp. C9.6.1-9.6.5, 1981.

[4] L. Renxiang, Z. Bing and M. L. Liou, "A New Three Step Search Algorithm for Block Motion Estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 4, No. 4, pp. 438-442, 1994.

[5] P. Lai-Man and M. Wing-Chung, "A Novel Four-Step Search Algorithm for Fast Block Motion Estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 6, No. 3, pp.313-317, 1996.

[6] C.Yuk-Ying and W. B. Neil, "Fast search block-matching motion estimation algorithm using FPGA," *Visual Communication and Image Processing 2000 Proc. SPIE*, Vol. 4067, pp.913-922, 2000.

[7] J. Jain and A. Jain, "Dispalcement measurement and its application in interframe image coding," *IEEE Transactions on Communications*, Vol. COM-29, pp. 1799–1808, 1981.

[8] C. Guy, G. Michael and K. Faouzi, "Efficient Motion Vector Estimation and Coding for H.263-based very low bit rate video compression," *ITU-T SG 16, Q15-A-45*, pp. 18, 1997.

[9] H. S. Yoon and G. S. Lee, "A modified Diamond search algorithm for fast block-matching motion estimation," *Proceeding of the 2001 korean Signal Processing Conference*, Vol. 14, No. 1, pp. 393-396, 2001.