Applied Mechanics and Materials
ISSN: 1662-7482, Vol. 855, pp 178-183
doi:10.4028/www.scientific.net/AMM.855.178

Submitted: 2016-06-30
Revised: 2016-08-17
Accepted: 2016-08-23
Online: 2016-10-25

A New Block Matching Algorithm for Motion Estimation

Chia-Ming Wu[1] and Jen-Yi Huang[2, a *]

[1]Taiwan Mao Chong Construction Co. Ltd, Taiwan

[2]Department of Computer Science and Information Engineering,
National Pingtung University, Taiwan

[a]huangjy@mail.nptu.edu.tw

Keywords: Motion Estimation, Block Matching Algorithms.

Abstract. Motion estimation has been the most key role in video processing. It is usually applied to block matching algorithm for choosing the best motion vector. The two adjacent images are searched to find the displacement of the same object in the video image. Many fast motion vector block matching algorithms are proposed, and they achieve the efficiency of motion compensation and video compression. In our paper, we propose a new algorithm that is based on ARPS. The experimental results show that the PSNR of the proposed method is better than that of other block matching methods on many kinds of video.

Introduction

With the growth of the Internet bandwidth and CPU performance, video compression has become a state-of-the-art technology and applied to the part of video processing. This method minimizes temporal and spatial redundancy to shorten file size but keeps the visual quality of the video.

MPEG-4 and H.264[1-2] are the most popular video compression standards in recent years. MPEG-4 is introduced in 1998 by the ISO/IEC Moving Pictures Expert Group(MPEG). It performs better than the former MPEG standards, such as MPEG and MPEG-2.H.264 in announced in 2003 by Joint Video Team, which is from ISO/IEC and ITU/T. It provides less file size than other standards like MPEG4, while the quality of the video is still maintained. MPEG-4 and H.264 are widely applied to many famous applications, such as Blu-Ray Discs, HDTV and YouTube.

Motion Estimation is the most time-consuming part of MPEG-4 and H.264. The most similar parts of the adjacent frames are selected to reduce the redundancy of video file. Block matching is one of the methods to select the most similar pairs. It chooses a fixed size of macro block from the search window of the previous frame that is most similar to the current macro block. The displacement between the current macroblock and selected macroblock id called motion vector, like Fig. 1.

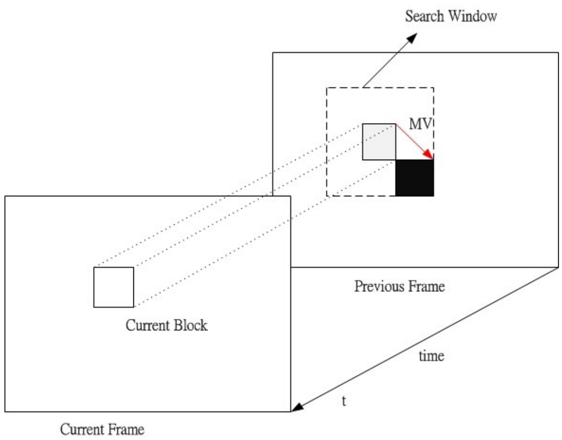

Fig. 1 Block Matching

Block Matching Algorithms. Full Search(FS) is the first and straight method for block matching algorithms. It searches all macro blocks in the search window to find the best solution. This scheme results in the most accurate solution, but it is also very time-consuming.

To decrease the computational complexity, many fast block matching algorithms are proposed, like Three-Step Search (TSS)[3], Cross Search (CS)[4], New Three-Step Search (NTSS)[5], Four-Step Search (FSS)[6], Diamond Search (DS)[7], Hexagon Search (HexaS)[8] and Adaptive Root Pattern Search (ARPS)[9].

The procedure of TSS is like Fig. 2. It searches 25 points to find the optimal solution.

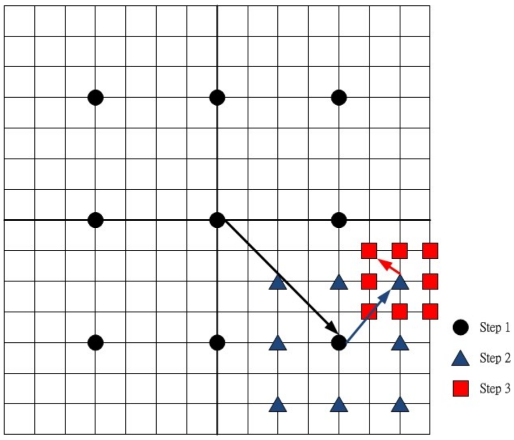

Fig. 2 Three-Step Search

CS is derived from TSS to decrease search points as Fig. 3.

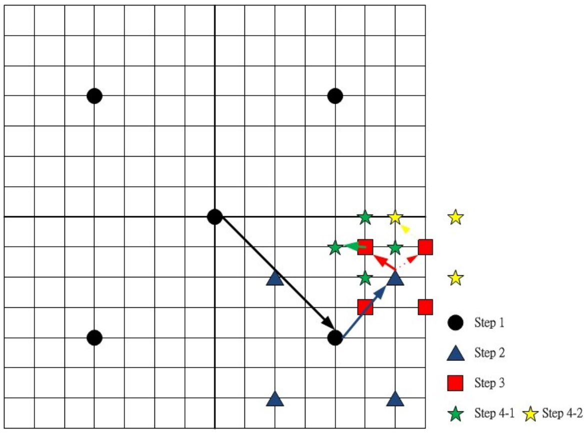

Fig. 3 Cross Search

NTSS is also obtained from TSS like Fig. 4, it applies early termination scheme to minimize the search points, especially slow and still motion videos.

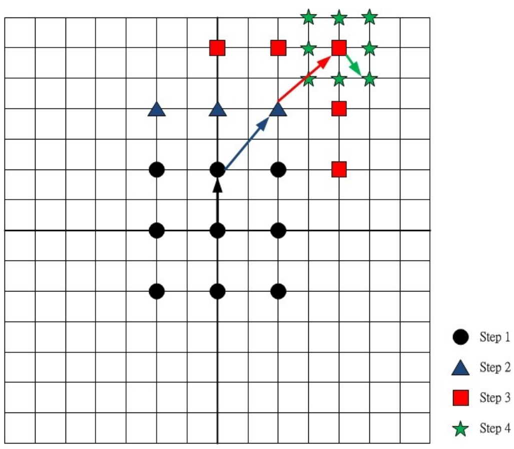

Fig. 4 New Three-Step Search

FSS is derived from NTSS to lower the search points of NTSS for both best and worst case as Fig. 5.

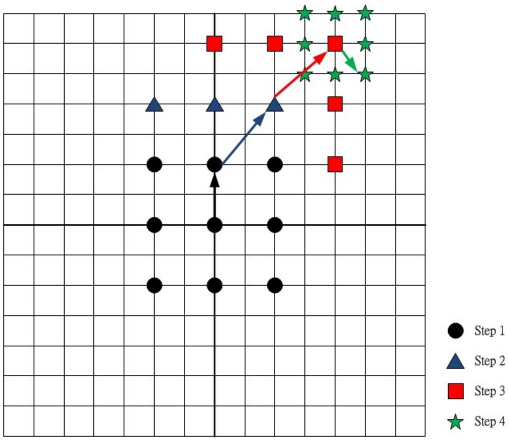

Fig. 5 Four-Step Search

DS reforms the inner search pattern of FSS to decrease the search points like Fig. 6.

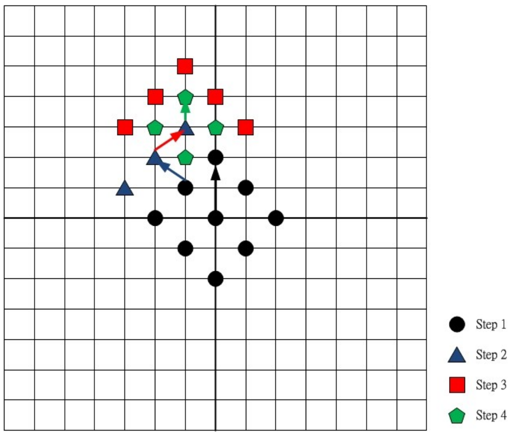

Fig. 6 Diamond Search

HexaS is obtained from DS as Fig. 7, the method modifies the search pattern od DS to reduce the search points of DS.

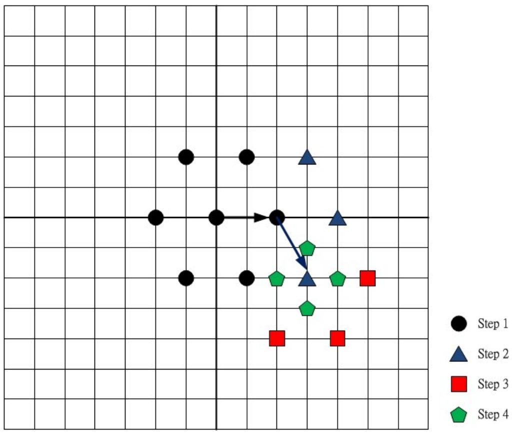

Fig. 7 Hexagon Search

ARPS is introduced in 2002 like Fig. 8, it determines the block size by the predicted vector to adapt different kinds of video. According to the statistics, ARPS uses 5 to 6 points to search the optimal motion vector in the best situation.

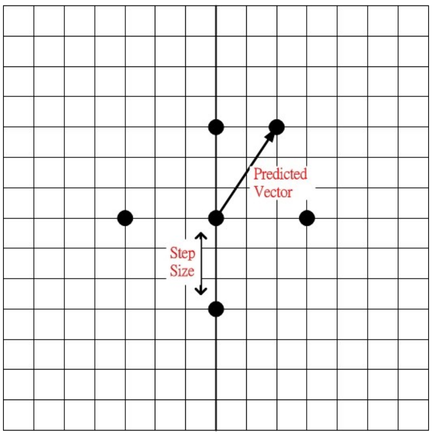

Fig. 8 Adaptive Root Pattern Search

The Proposed Method

Most of the non-zero motion vectors in the moving pictures are in horizontal-oriented. It means that lots of objects in the video are moving horizontally. So we adjust the pattern of ARPS by CS as Fig. 9. We call the proposed method as ARPS_F.

Differ from the first step of ARPS in Fig. 8, (stepSize, -1), (-stepSize, 1), (-stepSize, -1) and (0,0) are selected as the initial points. If the most suitable point is one of (-stepSize, -1), (0,0) and (stepSize, 1), blue pattern in Fig. 9 is chosen. Otherwise, the red pattern in Fig. 9 is selected to find the optimal solution.

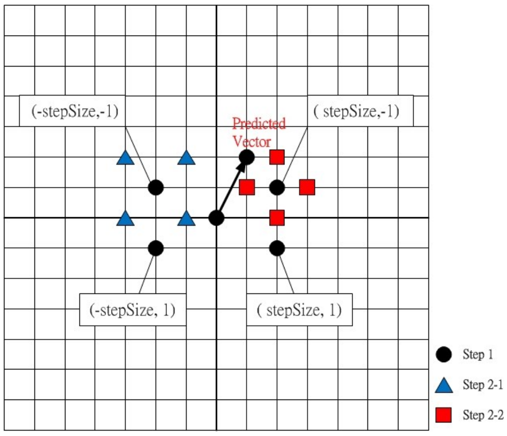

Fig. 9 The Proposed Method

Experimental Results

Ten different kinds of video chips are used as benchmarks of our experiment. In these video chips, highway(Qcif), stefan(cif), football(cif), bus(cif) and soccer(D1) belong to the high motion video chips, as akiyo and contain are slow or still motion video chips.

Table 1 Search Points for Qcif Videos

Search Points	akiyo	container	highway	carphone	foreman	suzie
FS	204.2828	204.2828	204.2828	204.2828	204.2828	204.2828
TSS	23.2121	23.2235	23.3561	23.3779	23.3715	23.2991
NTSS	15.9326	16.0571	18.6448	20.1565	21.6445	19.0216
CS	15.6147	15.6164	15.4878	15.5452	15.2060	15.4996
DS	12.2794	12.3766	14.7179	15.5740	17.2967	14.8406
HexaS	10.3503	10.4204	11.7099	12.1647	12.9730	11.7724
ARPS	5.0051	5.1263	7.4734	8.5178	9.1644	7.7128
ARPS_F	**5.0972**	**5.1451**	**8.0648**	**9.1428**	**9.6064**	**8.2291**

Table 2 Search Points for Cif and D1 Videos

Search Points	stefan	football	bus	soccer
FS	984.9192	984.9192	984.9192	1031.1
TSS	30.8109	31.0297	31.2964	32.1398
NTSS	26.7630	31.9689	30.7617	33.9839
CS	19.1128	18.8794	19.2424	19.4708
DS	19.8455	27.9194	22.5964	30.2482
HexaS	14.9968	19.3184	16.8777	20.6783
ARPS	10.2286	16.9483	11.6987	16.4677
ARPS_F	**10.4723**	**16.3464**	**11.9818**	**16.2531**

Table 1 and 2 show the result of search points. Our proposed method has the smallest search points than other methods in high motion and high-resolution movies.

Table 3 PSNR for Qcif Videos

PSNR	akiyo	container	highway	carphone	foreman	suzie
FS	41.8002	38.6291	35.0283	31.4563	30.7797	35.3929
TSS	41.3573	38.6119	33.8710	30.7606	29.7806	34.3532
NTSS	41.7784	38.6103	34.1808	31.0791	30.2027	35.1228
CS	39.3423	38.5918	31.3060	27.8715	26.1993	30.4841
DS	41.7639	38.5898	34.0890	30.9120	29.9650	34.8792
HexaS	41.1649	38.5932	33.4618	30.4203	29.3545	34.1450
ARPS	41.7451	38.5863	33.7233	30.7519	30.0512	34.9303
ARPS_F	**41.5861**	**38.5789**	**33.6361**	**30.5673**	**29.7565**	**34.3673**

Table 4 PSNR for Cif and D1 Videos

PSNR	stefan	football	bus	soccer
FS	24.5440	23.4189	23.7901	23.1183
TSS	22.8899	22.4764	21.3448	21.9441
NTSS	22.6536	22.3567	20.9349	21.9041
CS	18.5191	20.8555	17.7521	21.4805
DS	21.5715	21.6835	20.0665	21.2313
HexaS	21.3881	21.6061	19.9754	21.1084
ARPS	23.2543	22.1474	21.0764	22.1118
ARPS_F	**22.8443**	**21.9285**	**20.8073**	**21.7078**

Results in Table 3 and 4 demonstrate that our proposed method has the average PSNR.

Conclusions

In this paper, a fast block matching algorithm that is based on ARPS and CS is proposed. Experimental results show that it is suitable for high speed and high motion video chips. We will try to enhance the algorithm by keeping PSNR and minimizing search points to speed the video coding shortly.

References

[1] Joint Video Team of ISO/IEC and ITU-T, Advanced video coding for generic audiovisual services (ITU-T Rec. H.264 | ISO/IEC 201201-S AVC), 2012

[2] ISO?IEC, ISO/IEC 14496-1, 2001

[3] T. Koga, K. Linuma, A. Hirano, Y. Iijima and T.Ishiguro,"Motion-compensated interframe coding for video conferencing," Proceedings of NTC'81, pp.G5.3.1-G5.3.5, New Orleans, LA, Dec. 1981

[4] M. Ghanbari, "The Cross-Search Algorithm for Motion Estimation", IEEE Trans. on Communications, vol. COM-38, no. 7, pp.950-953, Jul. 1990

[5] Renxiang Li, Bing Zeng, and Ming L. Liou, “A New Three-Step Search Algorithm for Block Motion Estimation”, IEEE Trans. on Circuit System and Video Technology, vol.4, no. 4, pp.438-442, Aug. 1994

[6] L M. Po and W. C. Ma,”A novel four-step search algorithm for fast block motion estimation,”IEEE Trans. on Circuit System and Video Technology, vol. 6, pp.313–317, Jun. 1996

[7] J. Y. Tham, S. Ranganath, M. Ranganath, and A. A. Kassim, “A novel unrestricted center-biased diamond search algorithm for block motion estimation”, IEEE Trans. on Circuit System and Video Technology, vol. 8, no. 4, pp.369-377, Aug. 1998

[8] Ce Zhu, Xiao Lin, and Lap-Pui Chau, “Hexagon-Based Search Pattern for Fast Block Motion Estimation”, IEEE Trans.on Circuit System and Video Technology, vol. 12, no. 5, pp.349-355, May 2002

[9] Yao Nie, and Kai-Kuang Ma,“Adaptive Rood Pattern Search for FastBlock-Matching Motion Estimation”IEEE Trans on Image Processing, vol. 11, no. 12, pp.1442-1449, Dec. 2002