International Journal of advanced studies in Computer Science and Engineering
IJASCSE, Volume 3, Issue 11, 2014

A Fast Adaptive Rood Pattern Algorithm for Video Compression

Saeid Fazli
Research Institute of Modern Biological Techniques, University of Zanjan
Zanjan, Iran

Leila Tavakkoli
Dept of Electrical Engineering
University of Zanjan
Zanjan, Iran

Abstract— Video compression standards like H.26x/MPEGx cover almost all video applications such as mobile services, video conferencing, IPTV, HDTV, and HD video storage. Motion estimation has a significant role in digital video compression. In order to achieve the real-time property, a fast and adaptive algorithm is proposed. The adaptive rood pattern search (ARPS) is a rather useful way for estimating motions. This paper analyzes center-bias property and motion vector distribution length, and offers a new method based on ARPS. The proposed algorithm chooses adaptively different searching strategies as for the type of the motion vectors. The experimental results show that the proposed algorithm is 135.106 times faster than Full Search (FS) and 1.044 times faster than zero-motion prejudgment ARPS (ARPS-ZMP). This method also achieves a higher average peak signal-to-noise ratio (PSNR) of 31.504 compared to 31.39 of the ARPS-ZMP.

Keywords- Block-Based motion estimation (BMME), Adaptive rood pattern search (ARPS), video compression

I. INTRODUCTION

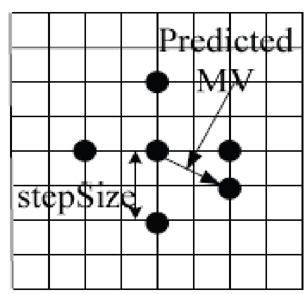
Motion estimation is one of the most important parts of video compression so that the progress of video coding is directly proportional to the progress of motion estimation. In recent years much progress has been made in this field, however, the motion estimation is the most important research field in the video compression. This effectively eliminates redundancy between frames and reducing the size of the video. In motion estimation, for several frames, a frame as reference is chosen. According to the reference frame, the rest are estimated.

Block matching is the usual method of motion estimation. In this way, the frame is divided into blocks and the each block displacement is calculated relative to the reference frame. There is different algorithms to calculate the displacement, such as Full Search (FS), Three-Step Search (TSS) [1], Diamond Search (DS) [2], Four-Step Search (4SS) [3], Simple and Efficient Search(SES) [4], ARPS [5], etc. The FS algorithm has best accuracy, but its computations cost is very high so that cause for rejecting this algorithm. Among these algorithms, ARPS, due to their higher speed and good accuracy, is mostly considered.

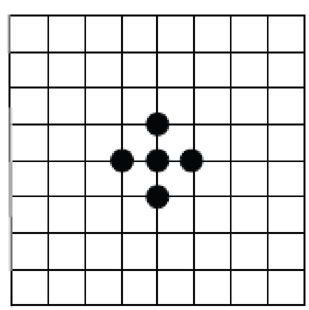
In this paper, Adaptive Rood Pattern Search algorithm (ARPS) is used to detect block transitions. Section II gives an introduction to Adaptive Rood Pattern Search; section III gives analysis of motion vector length. In section IV, the proposed search algorithm is explained. Section V presents experimental simulation and analysis. Section VI forms the conclusion.

II. ADAPTIVE ROOD PATTERN SEARCH

The most important feature of ARPS is coherent imagination of the motion in a frame. If the macro blocks around a current macro block are moving in a particular direction then there is a high possibility that the current macro block will also move in that direction. In this method, for motion prediction, left block is used. An example is shown in Figure 1.a. In this Fig the first step of rood pattern search is shown. It puts the search in a location where there is a high probability of finding a good matching block. Here, Step-Size = Max (|X|, |Y|) is used where X and Y are the x-coordinate and y-coordinate of the predicted motion vector.



(a)



(b)

Figure 1: a. Adaptive Root Pattern Search Method,
b. small rood pattern search (SRPS)

If predicated point is overlapping with one of the vertex points in the rood shape, it needs to search 5 points, otherwise 6 points, as shown in Figure 1.a. After searching these 5 or 6 points, a minimum cost point is obtained. This point is taken as the center point to go into the local search. ARPS uses a small rood pattern search (SRPS) for local searching, as shown in Figure 1.b. [6]

There are various cost functions, of which the most popular and less computationally expensive is Mean Absolute Difference (MAD) given by equation (1). Another cost function is Mean Squared Error (MSE) given by equation (2).and Sum of Absolute Difference (SAD) given by equation (3).

$$MAD = \frac{1}{N^2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\left|C_{ij} - R_{ij}\right| \quad (1)$$

$$MSE = \frac{1}{N^2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\left(C_{ij} - R_{ij}\right)^2 \quad (2)$$

$$\mathrm{SAD}(u, v) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\left|C(i+u, j+v) - R(i, j)\right| \quad (3)$$

Where N is the side of the macro bock, Cij and Rij are the pixels being compared in current macro block and reference macro block. Peak-Signal-to-Noise-Ratio (PSNR) given by equation (4) determines the motion compensated that is created by using motion vectors and macroblocks from the reference frame.

$$PSNR = 10Log_{10}\left[\frac{(Peak\ to\ peak\ value\ of\ original\ data)^2}{MSE}\right] \quad (4)$$

III. Analysis of Motion Vector Length

In order to improve search algorithm, it is necessary to investigate image distribution length of motion vectors. Table 1 shows the percent distribution probability in the different natural image sequence with FS algorithm where R is length of motion vector.

TABLE 1: DISTRIBUTION PROBABILITY PERCENTAGE OF MOTION VECTOR LENGTH

Videos	R =0	R <2	2<= R <4	4<= R	type
Akiyo	98.2256	1.4798	0.2896	0.0050	Low
Container	98.1397	1.3283	0.2239	0.3081	Low
Silence	89.4276	4.7559	2.4512	3.3653	Medium
Mother and daughter	84.8603	9.2761	3.4747	2.3889	Medium
Coastguard	64.9360	12.4125	14.6296	8.0219	High
Tennis(sif)	63.4929	7.4939	10.1990	18.8142	High
Coastguard(cif)	56.3119	8.6427	11.7908	23.2546	High
Flower(cif)	48.5970	13.8838	16.8444	20.6748	High

These sequences cover a wide range of motion contents and have various formats including QCIF, CIF, and SIF. Akiyo, Container, Silence, Mother and daughter and Coastguard has 300 frames with QCIF format. Tennis has 150 frames with SIF format. Coastguard (CIF) and Flower has respectively 300 and 250 frame with CIF format. This table also contains various types of motion speeds including low, medium and high.

Statistical data from Table 1 show that most video sequences contain either static micro block (MB) or moving with a radius of two around the center-point. This indicates that MVs of video sequence have center-bias property. As can be seen in TABLE 1, more than 62% of MVs are less than two.

.

IV. The Proposed Search Algorithm

While ARPS only adopts the left block to predict the current MV, The proposed algorithm makes full use of the center-bias property in horizontal and vertical direction so it employs left block in horizontal and top block in vertical direction to predict reference blocks. As mentioned in section III, most of the blocks are static and motion vectors have center-bias property. Therefore, zero-motion prejudgment [5] can be also employed to reduce computations.

Zero-motion prejudgment reduce the number of searches by predicting that if the block motion in next frame is zero and therefore it can skip the search for that block. To define static block, it compares block's SAD with a threshold T. First we compare SAD (0, 0) with T. If SAD(0,0) < T, it means that the current block is very similar to the corresponding block in the reference frame, then we can put the current block as a static block, set (0,0) as its MV, and finish search directly. Statistical experiment shows that more than 98 percent of static blocks SAD values are less than 512[7], so we take 512 as the threshold T.[8] Figure 2 shows the flow chart of the choosing Step-Size algorithm.

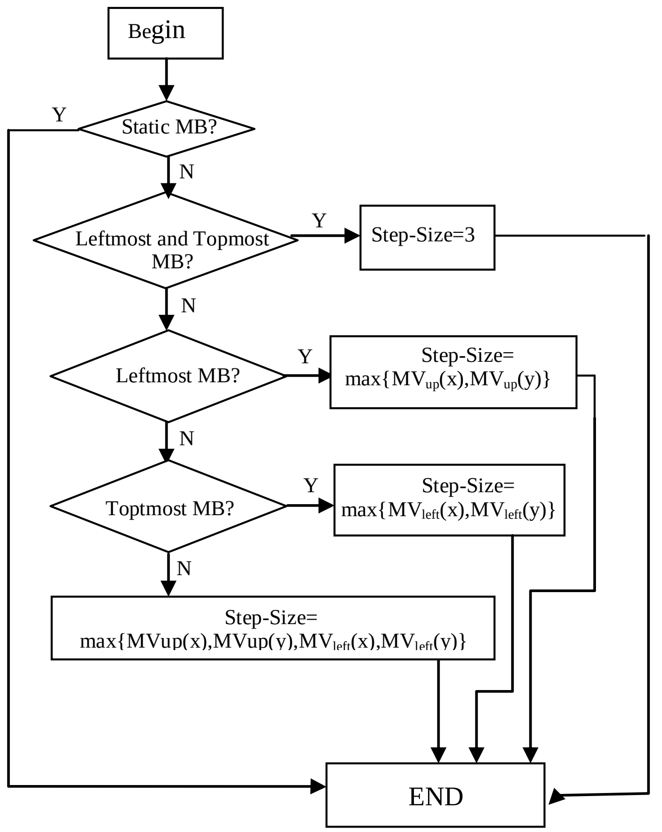


Figure 1. Flow chart of the choosing Step-Size algorithm

- **Step1:** Compute the matching error (SAD_{center}) between the current block and the block at the same location in the reference frame (i.e., the center of the current search window).

If SAD_{center} <T
  MV= [0 0];
  stop;
Else
  If the current block is a leftmost and topmost boundary block,
    Step-Size=3;
  Else
  If the current block is a leftmost boundary block,
    Step-Size=MAX{$|MV_{up}(x)|,|MV_{up}(y)|$};
  If the current block is a topmost boundary block,
    Step-Size=MAX{$|MV_{left}(x)|,|MV_{left}(y)|$};
  If the current block isn't leftmost nor topmost boundary block,
    Step-Size=
      MAX{$|MV_{left}(x)|,|MV_{left}(y)|,|MV_{up}(x)|,|MV_{up}(y)|$};
  go to **Step2**.

- **Step2:** if StepSize < 2 go to **Step4** else go to **Step3**
- **Step3:** Optimized ARPS algorithm. Start search from the original point at the corresponding block in the reference frame set StepSize as rood-shaped arm length. Compare the MAD value for the current block, set the point with minimum MAD value as initial search point, then go to step 4.
- **Step4:** SRPS algorithm. Set the point with minimum MAD value found in last step as the center in the small rood pattern, and check if the MAD value of center point is less than other points, go to step 5 directly, else, continue at step 4.
- **Step5:** Stop the algorithm.

V. EXPERIMENTAL SIMULATION AND ANALYSIS

In order to prove the availability of the proposed algorithm, we use the typical sequence which used in section III. FS, DS, ARPS and ARPS-ZMP [5] algorithms are adopted at the same experimental situations to compare with the proposed algorithm. Experiments are on the Matlab platform, and use macro block size of 16*16 pixels. Two indicators are used to compare the performance of each algorithm. One is the average searching points per block; the other is the average PSNR of compensated sequence. Average searching points per block in are used for comparison of the computational complexity, and the average PSNR is adopted for comparison of search precision. Experiment data in detail are shown in Table 2 and Table 3.

TABLE 2 : AVERAGE SEARCHING POINTS PER BLOCK

Video	FS	DS	ARPS	ARPS-ZMP	Proposed method
Akiyo	594	11.495	4.9743	1.4674	1.474
Container	594	11.4943	4.9677	1.5822	1.5592
Silence	594	12.4697	6.1133	3.839	3.2252
Mother and daughter	594	12.1903	5.8067	2.6575	2.6958
Coastguard	594	14.9697	7.863	7.059	6.883
Tennis(SIF)	652.34	16.5493	8.3313	6.5798	6.4514
Coastguard(CIF)	659.8	18.76	9.176	8.383	7.9667
Flower(CIF)	659.8	17.8016	9.1116	6.6272	6.3228
AVRAGE	**617.7425**	**14.46624**	**7.042988**	**4.774388**	**4.572263**

TABLE 3 : AVERAGE PSNR COMPARISON

video	FS	DS	ARPS	ARPS-ZMP	Proposed method
Akiyo	40.3850	40.3801	40.3776	40.3761	40.3803
Container	37.6927	37.6121	37.5833	37.5817	37.5858
Silence	32.1832	31.7645	31.6741	31.6741	31.7210
Mother and daughter	37.5942	37.5018	37.4693	37.4490	37.4811
Coastguard	28.7936	28.5473	28.6766	28.6745	28.7456
Tennis(SIF*150)	27.0864	25.6206	25.7706	25.7706	26.0945
Coastguard(CIF)	27.3882	26.4237	27.0584	27.0584	27.2457
Flower(CIF*250)	22.9439	21.3325	22.6102	22.5806	22.7786
AVRAGE	**31.7584**	**31.14783**	**31.40251**	**31.39563**	**31.50408**

Table 2 shows that in the sequence with slow motion like Akiyo, this algorithm increases 402.98 times in search speed, compared with the FS algorithm .It also obviously speeds up in the medium motion and large motion sequence so that the average of searching points is 135.106 times faster than Full Search (FS) and 1.044 times faster than ARPS-ZMP. Moreover, the algorithm not only finds out the optimized MV very fast, but also keeps high search accuracy at the same time. The proposed method achieves an average peak signal-to-noise ratio (PSNR) of 31.504 which is higher than the one of ARPS-ZMP method (31.39) as shown in TABLE 3.

VI. CONCLUSION

In order to improve the search speed and the search accuracy of motion estimation in image sequence, the distribution length of image motion vector is analyzed firstly. Based on this analysis, we suggest a new algorithm that finds out the optimized MV quickly and keeps high search accuracy at the same time. Experimental results show that this algorithm meets the real-time demand, and it is quick, effective and easy for implementation.

REFERENCES

[1] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, “Motion compensated interframe coding for video conferencing,” in Proc. Nat. Telecommunication Conf., Nov. 29–Dec. 3, 1981, pp. G5.3.1–G5.3.5.

[2] S. Zhu and K. K. Ma, “A new diamond search algorithm for fast block-matching motion estimation,” IEEE Trans. ImageProcessing, vol. 9, no. 2, Feb. 2000,pp.287-290

[3] 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. 313–317, June 1996.

[4] Jianhua Lu, and Ming L. Liou, “A Simple and Efficent Search Algorithm for Block-Matching Motion Estimation”, IEEE Trans. Circuits And Systems For Video Technology, vol 7, no. 2, pp. 429-433, April 1997.

[5] Yao Nie, and Kai-Kuang Ma," Adaptive Rood Pattern Search for Fast Block-Matching Motion Estimation", IEEE Trans. Image Processing, vol 11, no. 12, pp. 1442-1448, December 2002.

[6] Vanshree Verma, Sr. Asst. Prof. Ravi Mishra," A Novel Adaptive Rood Pattern Search Algorithm", IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE).e-ISSN: 2278-1676 Volume 4, Issue 4 (Jan. - Feb. 2013), PP 14-18M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

[7] Hosur P. I, K. K. Ma, “Motion Vector Field Adaptive Fast Motion Estimation, ” International Conf. Information, Communication and Signal Processing’99, Singapore ci ty, Singapore, 1999:7-10

[8] Mu-Long Lin, Qing-Ming Yi, Min Shi," A Fast and Adaptive Search Algorithm Based on Rood Pattern and Gradient Descent",Physics Procedia 33 (2012) 1526 – 1