Architecture For Adaptive Rood Pattern Search Algorithm For Motion Estimation

K.Leela Bhavani
Electronics & Communication Engineering Department,
V.K.R., V.N.B. & A.G.K. College of Engineering,
Gudivada, India

R.Trinadh
Electronics & Communication Engineering Department,
C.R.Reddy College of Engineering,
Eluru, India.

Abstract— **In this report a simple and novel fast Block Matching Algorithm (BMA), called Adaptive Rood Pattern Search (ARPS) is being presented. The Algorithm contains two sequential search stages:1) Initial Search and 2)Refined Local Search. For each Macro Block (MB), the initial search is performed once at the beginning in order to find a good starting point for the follow-up refined local search. By doing so unnecessary intermediate searches and the risk of being trapped into local minimum matching error can be reduced in the long search case. For the Initial Search stage an adaptive rood pattern (ARP) is proposed, and ARP size is dynamically determined from each MB, based on the available motion vectors (MVs) of the neighboring MBs. In the refined local search stage a unit rood pattern size is exploited repeatedly. The flow of this report starts with the introduction, which will be followed by the detailed explanation of algorithm and then finally ending with MATLAB simulation results, proposed architecture, conclusions and future work.**

Keywords—**ARP,MB,MV,BMA,ROS**

I. INTRODUCTION

Ever since the development of video compression algorithms, there was a much concern about the motion estimation algorithms, because they are the one’s which help in achieving the video compression to the most. As the time past cardinal Motion Estimation algorithms came into existence of which one class of algorithms called Block Matching Algorithms(BMA) has been widely adopted by current video coding standards such as H.264, H.261, H.263, MPEG-1, MPEG-2, MPEG-4 due to its effectiveness and simplicity of implementation.

The most straightforward BMA is the full search (FS) which exhaustively searches for the best matching block within the search window. However FS yields a very high computational complexity and makes ME the main bottle neck in the real time video coding applications. Thus using a fast BMA is indispensible to reduce the computational cost. Among the fast BMA algorithms the most popular ones are 1) 3 Step Search (3SS) 2) 4 Step Search (4SS) 3) Diamond Search (DS). The various variants of the above mentioned searches also exists such as New 3SS, Large DS, Small DS etc., Apart from the above mentioned fast BMAs one more recently developed search was Adaptive Rood Pattern Search(ARPS). This report mainly aims in proving ARPS as better over Diamond search, Gradient block search, Full Search in terms of Number of search points, Number of computations, but maintaining PSNR values in close to the FS,DS and better over Gradient block search. The simulations were performed on MATLAB 7.2. This report also gives information regarding the proposed architecture. The HDL used for the architecture development is VERILOG and the synthesis tool used is the Xilinx ISE version 9.2.

II. ADAPTIVE ROOD PATTERN SEARCH

A) Overview

The Speed and the accuracy of the motion estimation algorithms depends on the size of the search pattern and the magnitude of the target MV, as the small search patterns are useful in detecting small motions but they tend to trap into the local minimum while detecting the large motions, on the other hand the large motion vectors can easily detect the large motions but they tend to go for unnecessary searches when detecting the small motions. Hence it is desirable to use different search patterns according to the estimated motion behavior (in terms of the magnitude of motion) for the current block. This boils down to two issues required to be addressed: 1) How to predetermine the motion behavior of the current block for performing efficient ME? and 2) What is the most suitable size and the shape of the search pattern(s)?

Regarding the first issue, in most cases adjacent MBs belonging to the same moving object have similar motions. Therefore the motion vector for the current MB cab be reasonably predicted from the neighboring MVs in the spatial or temporal domains. As for the second issue we use two types of search patterns. One is adaptive rood pattern (ARP) with adjustable rood arm (thus, pattern size), which is dynamically determined for each MB according to the predicted motion behavior. Note that the ARP will be exploited only once in the beginning of the MB search. The objective is to find a good starting point for the remaining local search so as to avoid unnecessary intermediate search and reduce the risk of being trapped into the local minimum in case of long search path. The starting point identified is hopefully as close as to the global minimum as possible. If so then, a small fixed size search pattern will be able to complete the remaining local search quickly. Note also that this small search pattern will be used repeatedly unrestrictedly until the final MV is found.

B) Prediction of the Target MV

In order to obtain the accurate MV prediction of the current block two factors need to be considered: 1) Choice of the Region Of Support (ROS) that consists of the neigh boring blocks whose MVs are used to calculate the predicted MV, and 2) algorithm used to construct the predicted MV.

In the temporal region the block in the reference frame at the same position as that of current block in the present frame is a straight forward choice as a temporal ROS candidate. However, the neigh boring blocks from the same reference frame can also be used for prediction. However there would be a large requirement of memory if such a kind of operation is performed, as the MV information of the complete reference frame should be stored. So the choice of temporal prediction will be eliminated due to the huge memory requirement and computations. The other way possible is to go for the spatial prediction. Usage of the already calculated i.e the neigh boring blocks MVs as a source for prediction will be a good option. It is the only possible way to have less memory requirement. The concept of Region of Support(ROS) is used for the prediction of current block MV. There are 4 kinds of ROS possible. They are as follows.

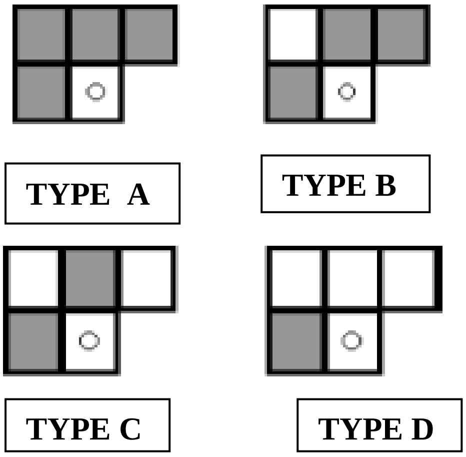


Fig. 1 Types Of Region Of Supports

TYPE A ROS covers all the four neighboring blocks and TYPE B is the prediction ROS that is adopted in some international standards such as H.263 for the differential coding of the MVs. TYPE C composed of the two directly adjacent blocks, TYPE D consists of only one adjacent block that is left of the current MB. Experiments on various types of ROCs is being done and it was observed that they yield fairly similar results with a difference of less than 0.1 DB in PSNR and 5% in the number of search points. Hence it is wise to choose TYPE D kind of ROS hence it requires only one motion vector for prediction.

C) Selection Of Search Patterns

1) Adaptive Rood Pattern- For initial Search: The shape of the rood pattern is symmetrical that is shown in the figure 2. The main structure of ARP takes the rood shape, its size refers to the distance between center point and the any of the other vertex point. The shape of the rood shape is determined on the basis of real world motion sequences. For most of the sequences it was observed that the motion vector distribution was mostly in horizontal and vertical direction than in other directions, since the camera movements are mostly in those direction. Since the Rood pattern spreads in both the Vertical and Horizontal directions it can quickly detect the motion vectors and also can able to jump directly into the local region of the global minimum. Secondly, any MV can be decomposed into one vertical MV component and one horizontal MV component. For a moving object which may introduce motion in any direction the rood shaped pattern can atleast detect the major trend of the moving object which is the desired outcome of the initial search stage. Furthermore ARP's Symmetric shape is advantageous in terms of hardware implementation.

In addition to the four search point it would be better to include the position of the predicted motion vector, that aids in the termination in the initial search stage only if the predicted MV matches with the target MV.So in total there will be 6 search points in the initial search stage and then 5 points for the further refinement process. The search pattern that will be used in the initial search stage is shown in the fig 2.

In this method the Rood arm length will be equal to the size of the predicted motion vector for the initial search stage, and the four arms are of equal length. Mathematically it can be expressed as follows. The size of the ARP, Ѓ is

Ѓ = *Round* MV predicted

$$= [\sqrt{(MV^2_{predicted}(x) + MV^2_{predicted}(y))}]$$

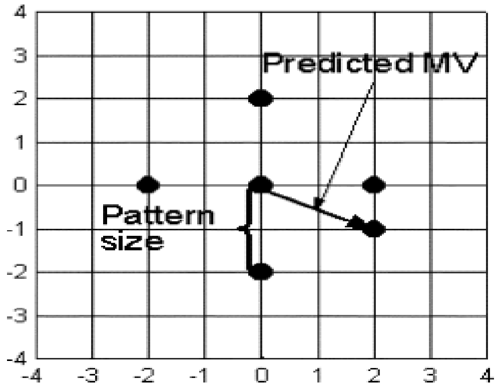


Fig. 2 Adaptive Rood Pattern

Where the $MV_{predicted}(x)$ is the x-component of the predicted motion vector and the $MV_{predicted}(y)$ is the y-component of the predicted motion vector. Operator round performs the rounding operation to the nearest possible integer since the displacement can be in terms of the integers.

The above equation involves quadratic computation, which is a time consuming process. In terms of hardware implementation it is suggested to not include square root kind of operations. In this concept of motion estimation we need to speed up the process as far as possible so that we can meet the desired frame rate. Instead of the Square root operation, in order to maintain the hardware complexity less the greater of the magnitudes of the x and y components taken as the size of the Rood arm. That is

$Ѓ = \text{Max}(|MV_{predicted}(x)|,| MV_{predicted}(y)|)$.

It should be kept in mind that the TYPE D kind of ROS cannot be used for all MBs, because there will be some MBs where there will be no left neighbour. So in that case the ARP with rood arm length of 2 ($Ѓ = 2$) is suggested, by taking the reference of Large Diamond Search Pattern, which has fairly a good amount of performance. Also larger MVs are not preferred as the boundary MVs mostly belong to the static background, which do not contribute larger MVs.

2) Fixed Pattern – For Refined Local Search: In the initial search the adaptive rood pattern directly leads to the new search position which is somewhere around global minimum, which avoids the unnecessary search points in the intermediary search path. Since there is no chance of getting trapped into the local minimum we can use the fixed pattern for identifying the global minimum. The minimum error point in the first step is used to align as the centre of the fixed pattern in the second step. This process will be followed until the point of minimum error is the centre of the present iteration's search pattern. Two types of fixed patterns were proposed. The first one was the 3x3 square pattern as was proposed in the Small Diamond Search Pattern (SDSP). The second pattern consists of a Unit size rood arm pattern. The experimental results conducted by [1] showed that the 3x3 square pattern yields similar PSNR when compared to the Unit rood arm pattern but 40% to 80% more number of search points. This demonstrates the efficiency of the Unit Rood Arm Pattern. The proposed fixed patterns are shown in the fig 3.

D) Summary of the ARPS method

Step 1:- Compute the matching error(SADcentre) between the current block and the block at the same location in the reference frame(i.e centre of the current search window).
If the current block is the left most
$Ѓ = 2$;
Else
$Ѓ = \text{Max}(|MV_{predicted}(x)|,| MV_{predicted}(y)|)$
Go to step 2

Step 2:- Align the centre of ARP with the centre point of the search window and check its 4 points and the position of the predicted motion vector to find the minimum error point.

Step3:- Set the centre point of the unit size rood pattern at the minimum error point found in the previous step and check its points. If the new minimum error point is not incurred at the centre of the unit rood pattern repeat this step otherwise, MV is found corresponding to the minimum error point in the current step.

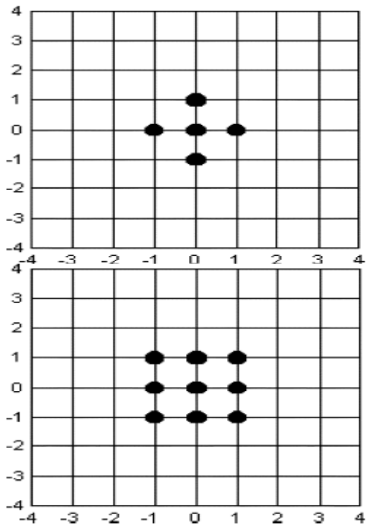


Fig 3:- Proposed Fixed size Patterns

In this implementation of the algorithm a flag is maintained which takes care of the already computed search points, so that redundant computations can be avoided and hence speeding up the process.

III) EXPERIMENTAL RESULTS

Simulation of this algorithm has been performed using the MATLAB 7.2 version software. I did the comparison of ARPS with the 3SS and the FS algorithms, in terms of the PSNR, number of search points and the number of SAD computations. The test frames that were applied were of HDTV frames of 720p and 1080p resolution. The simulation was performed by considering one starting original frame and a series of 14 frames were predicted using the first original frame.

AVARAGE PSNR(in DB)

Video Name	**ARPS**	**DS**	**GDS**	**FS**
Stock Holm	30.54	31.20	29.8	31.3
MobCal	32.85	33.43	33.45	33.46
Sheilds	27.56	28.35	25.35	29.31
ParkRun	19.44	21.81	19.20	20.91

AVARAGE NUMBER OF SEARCH POINTS PER BLOCK

Video Name	**ARPS**	**DS**	**GDS**	**FS**
Stock Holm	10.56	12.2	15.8	225
MobCal	6.40	9.25	9.63	225
Sheilds	15.40	14.26	19.85	225
ParkRun	10.53	12.07	15.45	225

IV) ARCHITECTURE

1) *Principle Of Belongingness:*

The proposed architecture is primarily based on the principle of belongingness. The principle of belongingness can be stated as: The pixel belongs to a particular search point if its position is within the boundary of the search point, where boundary size is equal to the size of the macro block.

For every pixel this belongingness is exploited w.r.t every search point and a header is added to the pixel. The information regarding the belongingness will be present in the header. Here 6 bit header is added to the pixel information and this 14 bit data is broadcasted to all the processing elements (PEs).Each bit in the header corresponds to the search point. As a maximum of 6 search points will be there in any iteration, 6 bit header is used in the proposed architecture. Now the 14 bit information is broadcasted to 6 PEs and a checker will be present before the PE, it will check whether a particular pixel will belong to the particular search point (SP) for which the PE is mainly concerned about. If it belongs then the data is forwarded and the corresponding current block(CB) information is extracted from the FIFO, which is followed by the calculation of Sum of Absolute

2) *Architectural aspects*

The proposed architecture is divided into 5 sections.

a) Search Area(SA) section
b) Header generation section
c) Processing Element(PE) section
d) Current block(CB) section
e) Controller section

The following sections will be explaining each and every section in detail.

a)Search Area section

In the proposed architecture, SA for the CB is brought into the on chip memory dedicated to the SA as and when required. The concept of Half search area (HSA)[] is used in the proposed architecture. Tha main aim of the concept of HSA was to reduce the I/O bandwidth and utilize the overlap between the search areas of adjacent CBs(see fig 4).

An HSA buffer stores an HSA block(30x16) which is larger than real half search area(30x15). The extra column is for filling the gap between search areas of different tasks. Fig 5 illustrates the operations of the three HAS buffers. When executing task 0(matching current block 0), PE's access search pixels from HAS 0 and 1.

Current blocks(16x 16) HSA blocks(30x16)

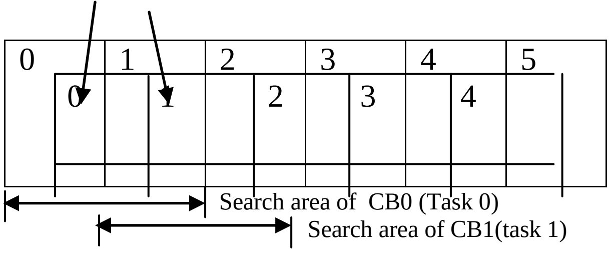


Fig. 4 Search areas of adjacent current blocks.

	T0	T1	T2	T3	T4	T5	T6
HSA Buffer0	0	**3**	3	3	**6**	6	6
HSA Buffer 1	1	1	**4**	4	4	**7**	7
HSA Buffer 2	**2**	2	2	**5**	5	5	**8**

Fig.5 Contents of the three half search area buffers (write operations are marked by small frames)

During these 256x3 clock cycles, the HAS buffer 2 is being filled by the right half of the search area for task 1(HAS block 2). In the next 256x3 clock cycles, search area pixels are accessed from the HAS buffers 1 and 2 for executing task 1,
and new data are input to buffer 0.

b) Header generation section:

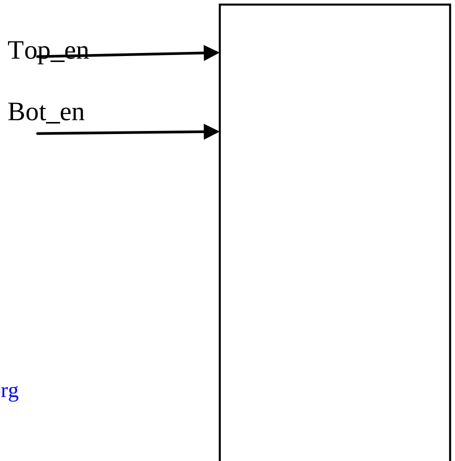
The main aim of this section is to exploit the belongingness and set the appropriate fields in the header. The structure of the header along with the pixel information will be as follows.

Header	Pixel Value

The task of header generation is accomplished by using Bitset array, which will develop boundaries and checks to see whether a pixel falls into the boundaries of any of the search points based on its position. For example, if a search point is within the boundaries of 2 search points top and bottom then the corresponding bit fields of the top and bottom are set. There is a possibility that the SPs can go outside the search window, in such a case the calculation with respect to those SPs should be disabled which is done by the 5 signals Top_en, Bot_en, Left_en, Right_en, Center_en. The inputs Top_en, Bot_en, Left_en, Right_en, Center_en along with the checking of belongingness used to set the bit fields in the header.

c) PE Section:

A PE is dedicated for each search point. As there are 6 search points at a maximum in a single iteration, maximum of 6 PEs are required. A checker is present in between the bit set array and the PE, so if a broadcasted pixel falls within the boundaries of the SP, to which the PE is concerned then the checker will latch the pixel information and sends a FIFO_en



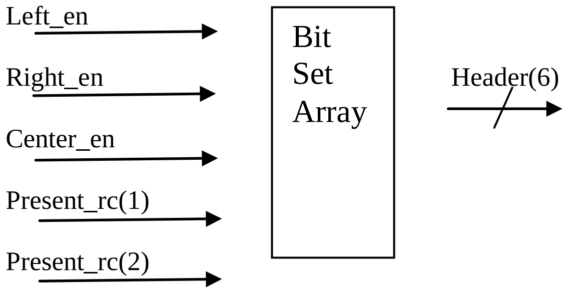


Fig. 6 Pin diagram of Bit set array structure.

signal to the corresponding FIFO in the CB section, which makes the corresponding CB data to get latched into a register. Then the PE element will do the required SAD calculation. At the end of the pixel in the search window when the final SAD calculation is done, the output's are fed to the minimum position finder block which finds the minimum value and gives the minimum position which is useful for further processing.

d) CB Section

Here an on chip memory for accessing the CB data. This CB data is copied into the FIFO's sequentially pixel by pixel. Whenever the FIFO_en from checker gets activated the FIFO_en signal from the checkergets activated then the FIFO data gets latched into the register which is further fed into the PE.

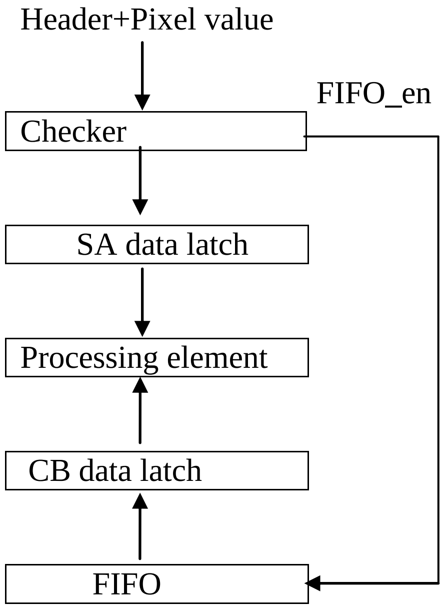


Fig. 7 Data flow for the computation of the SAD

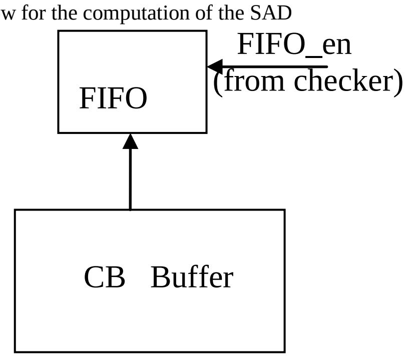


Fig. 8 Block diagram of CB Section

e) Controller section

The controller generates necessary control signals to make all the blocks work synchronously. This is achieved by a FSM. The controller even has a frame boundary checker which is used to check whether a search point is within the search window.

V) CONCLUSIONS

From the experimental results it can be concluded that the ARPS is superior over Gradient descent search in performance aspects as well as in the number of search points,and fairly close in performance with respect to DS and FS and as well as faster than DS and FS. The main aspect of ARPS that makes it superior over GDS in performance and faster than DS and FS is its adaptability nature of rood arm, because if the current MB belongs to the same object to which its neighbouring block belongs, with help of predicted MV, the search can be terminated within the initial search stage itself. If we look on the other side of this aspect, that if current MB does not belong to the same object as its neighboring block belongs to then also the patterns search points in horizontal and vertical direction will be able to track the minimum errored position as it was observed by [1] that most of the motion will be in the horizontal and vertical directions.

VI) REFERENCES

[1] "Adaptive Rood Pattern Search for fast block matching algorithms" by Yao Nie, Kai-kuang Ma, IEEE transactions on IMAGE PROCESSING, vol 11 , NO. 12, page no:- 1442-1449, DECEMBER 2002.

[2] "An enhanced Adaptive Rood Pattern Search Algorithm for Fast block matching motion estimation" by Hui Zhao, Xin bo Yu, Jia Hong Sun, ChangSun, Hao zhe Cong, 2008 IEEE Congress on Image and Signal Processing, page no:- 416-420.

[3] "An Adaptive Rood Pattern Search for Fast Block Matching Motion Estimation in JVT/H.26L" by Kai Kuang Ma and Gang Qiu, IEEE International Conference on Image Processing 2003,page no:- ii-708 – ii-711.

[4]"VLSI architectures for video compression - A Survey" by Peter Pirsch, Proceedings Of the IEEE, Vol 83, NO 2, February 1995.

[5] "A high throughput and low cost diamond search architecture for HDTV motion estimation" by Marcelo Porto, Luciano Agostini, Sergio Bampi, Altamiro Susin , IEEE International Conference on Multimedia and Expo 2008, pg no:- 1033-1036

[6] "An efficient VLSI architecture for Diamond Search Pattern based Motion Estimation algorithms" by Kin-Hung Lam and Chi-Ying Tsui, *Proc. of 2000 Asia Pacific Conference on Multimedia Technology and Applications (APCMTA2000),* Kaohsiung, Taiwan, Dec, 2000.

[7] "New VLSI architecture for motion estimation algorithm" by V.S.K Reddy, S. Senguptha, and Y.M.Latha , World academy of science, Engineering and technology 36 2007.

[8] "Parallel Architectures for 3-Step Hierarchial Search Block Matching Algorithm" by Her-Ming Jong, Liang-Gee Chen,Tzi-Dar Chiueh, IEEE transactions on circuits and systems for video technology, vol 4, no. 4, August 1994.

[9] "The Hardware architecture of a novel motion estimator with adaptive crossed quarter polar search patterns for H.264 encoding". by Yifeng Qiu and Wael Badawy, Proceedings of the 22nd Canadian Conference on Electrical and Computer Engineering, CCECE 2009, 3-6 May 2009, Delta St. John's Hotel and Conference Centre, St. John's, Newfoundland, Canada. IEEE 2009, pg no:-819 – 822.

[10] "A parallel Architecture for Successive elimination Block Matching Algorithm" by B.K.N.Srinivasarao, Indrajt Chakrabarti, Sixth Indian Conference on Computer Vision, Graphics & Image Processing, ICVGIP 2008, Bhubaneswar, India, 16-19 December 2008. IEEE 2008, pg no:- 226-231