IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 1, Issue 1, March, 2013
**ISSN: 2320 - 8791**
**www.ijreat.org**

# Cost Effective Motion Estimation Processors Design for Advanced Video Coding

[1]G.Gopinath, [2]J.Manikandan

[1]*Dept. of Embedded system technologies, Sri Sairam Engineering college, Chennai, India*

[2]*Dept. of ECE, Sri Sairam Engineering college, Chennai, India*

*Abstract*

**Motion estimation is a crucial step in today's video coding and processing systems. It is always regarded as the most time consuming module in video coding, and many fast motion estimation algorithms have been proposed to speed-up the video encoding. This paper proposes an adaptive fractional pel search algorithm based on H.264 to decrease the encoding time and to improve the encoding efficiency. The proposed Medium Biased Fractional Pel Search (MBFPS) algorithm improves the efficiency by eliminating the half pel search using the median of the surrounding pels. The proposed circuit has been designed at Register Transfer Level (RTL) level by using the VHDL language. The most complicated block in the H.264 architecture is the motion estimation and compensation block. It has the advantage of gaining in the computation time and memory space for recording images, and even the advantage of keeping the same memory addresses for the different blocks of object. The architecture proposition also serves to read block images of a video in real time. Compared with the fractional pel full search algorithm, the MBFPS algorithm shows that it can reduce 76.60% and 68.22% fractional pixel search points respectively while it still maintains similar coding efficiency.**

***Keywords— Motion Estimation, Motion vector, H.264 codec, Motion compensation.***

## I. INTRODUCTION

Innovations in the communication systems have been tremendous in the last decade. Technology in communication systems has transformed from having only analog television via cable, satellite with availability of only few channels or mobile phones that can only make voice calls or send SMS messages or internet connections that are slow, mostly connected through dial –up modem connected via telephone lines, back then data was stored on floppy disks, magnetic tapes and bulky hard drives.Today, video coding lies at the core of multimedia. Among the many technologies involved, video coding and its standardization are definitely the key enablers of these development.A typical video communication system, which relays video over a transmission channel, is shown in Figure 1.

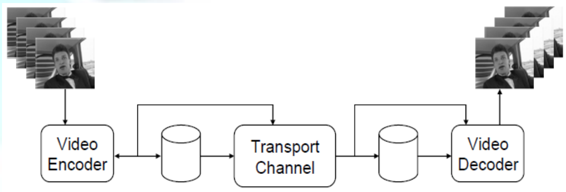


Fig 1. A Typical Video Communication System

Digital television sets have become more compact with availability of regional and international channels with HD quality. Data is stored on re-writable DVDs, Blu-ray discs and hard disks which are light weight, portable with huge space for storage. Internet connection is blazing fast with wireless routers and modems operating at faster speeds. In this fast growing world of communications, data compression is still one of the most essential components in any multimedia system. Modern data compression techniques offer the possibility to store or transmit the vast amounts of data necessary to represent digital videos and images in an efficient and robust way.

Compression is the process of removing redundant information and representing data with fewer bits than the original information would use. It is useful because it helps to reduce the consumption of expensive resources such as data storage on hard disks/servers and transmission bandwidths. Hence, research is still going on compression techniques to enable real-time data transmission using less resources. Compression techniques are categorized as lossless or lossy.

This thesis improves the efficiency of the coding techinque using Median Biased Fractional Pixel Search(MBFPS) which eliminates the half pixel search present in Herirarchical Fractional Pixel Search Algorithm(HFPS). The algorithm of MBFPS has been discussed.

Jose L. nunez-yanez *et al*. [1] has proposed a flexible and scalable motion estimation processor capable of supporting the processing requirements for high-definition (HD) video using the H.264 Advanced Video Codec, which is suited for FPGA implementation. Unlike most previous work, our core is optimized to execute all existing fast block matching algorithms, which we show to match or exceed the inter-frame prediction performance of traditional full-search approaches at the HD resolutions commonly in use today. Using our development tools, such algorithms can be described using a C-style syntax which is compiled into our custom instruction set. We show that different HD sequences exhibit different characteristics which necessitate a flexible and configurable solution when targeting embedded applications.

Jörn Ostermann *et at.* [2] provides an overview of the new tools, features and complexity of H.264/AVC H.264/AVC, the result of the collaboration between the ISO/IEC Moving Picture Experts Group and the ITU-T Video Coding Experts Group, is the latest standard for video coding. The goals of this standardization effort were enhanced compression efficiency, network friendly video representation for interactive (video telephony) non–interactive applications (broadcast, streaming, storage, video on demand). H.264/AVC provides gains in compression efficiency of up to 50% over a wide range of bit rates and video resolutions compared to previous standards. Compared to previous standards, the decoder complexity is about four times that of MPEG-2 and two times that of MPEG-4 Visual Simple Profile.

E. Hung *at et.*[3] presented a programmable, configurable motion estimation for the H.264 Video coding standard. The motion estimation is capable of handling the processing requirements of the process follows the ASIP(Application Specific Instruction Set Processor) approach with a instruction set algorithms. Configurability relates to the ability to optimize the micro architecture for the selected algorithm and performance requirements through varying the number and the type of execution units at compile time.

## II. Advanced Video Coding

The latest video compression standard, H.264 (also known as MPEG-4 Part 10/AVC for Advanced Video Coding), is expected to become the video standard of choice in the coming years.H.264 is an open, licensed standard that supports the most efficient video compression techniques available today. Without compromising image quality, an H.264 encoder can reduce the size of a digital video file by more than 80% compared with the Motion JPEG format and as much as 50% more than with the MPEG-4 Part 2 standard. This means that much less network bandwidth and storage space are required for a video file.

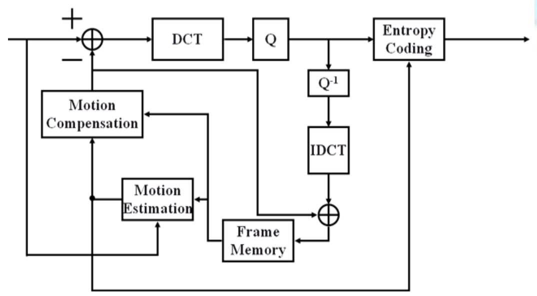


FIGURE 2 VIDEO ENCODER BLOCK DIAGRAM

### *A.DEVELOPMENT OF H.264*

H.264 is the result of a joint project between the ITU-T's Video Coding Experts Group and the ISO/IEC Moving Picture Experts Group (MPEG). ITU-T is the sector that coordinates telecommunications standards on behalf of the International Telecommunication Union. ISO stands for International Organization for Standardization and IEC stands for International Electro technical Commission, which oversees standards for all electrical, electronic and related technologies.

### B.H.264 PROFILES AND LEVELS

Depending on the H.264 profile, different types of frames such as I-frames, P-frames and B-frames, may be used by an encoder. An I-frame, or intra frame, is a self-contained frame that can be independently decoded without any reference to other images. The first image in a video sequence is always an I-frame. I-frames are needed as starting points for new viewers or resynchronization points if the transmitted bit stream is damaged.

I-frames can be used to implement fast-forward, rewind and other random access functions. An encoder will automatically insert I-frames at regular intervals or on demand if new clients are expected to join in viewing a stream. The drawback of I-frames is that they consume much more bits, but on the other hand, they do not generate many artifacts.

A P-frame, which stands for predictive inter frame, makes references to parts of earlier I and/or P-frame(s) to code the frame. P-frames usually require fewer bits than I-frames, but a drawback is that they are very sensitive to transmission errors because of the complex dependency on earlier P and I reference frames. A B-frame, or bi-predictive inter frame, is a frame that makes references to both an earlier reference frame and a future frame.


Fig 3. Typical Frame Sequence

A typical sequence with I-, B- and P-frames. A P-frame may only reference preceding I- or P-frames, while a B-frame may reference both preceding and succeeding I- or P-frames. In the H.264 baseline profile, only I- and P-frames are used. This profile is ideal for network cameras and video encoders since low latency is achieved because B-frames are not used.

### C.METHOD OF REDUCING DATA

Within an image frame, data can be reduced simply by removing unnecessary information, which will have an impact on the image resolution. In a series of frames, video data can be reduced by such methods as difference coding, which is used by most video compression standards including H.264. In difference coding, a frame is compared with a reference frame (i.e. earlier I- or P-frame) and only pixels that have changed with respect to the reference frame are coded. In this way, the number of pixel values that are coded and sent is reduced.

Fig 4 Motion JPEG Coding Sequence

The three images in the Figure 4 are coded and sent as separate unique images (I-frames) with no dependencies on each other.

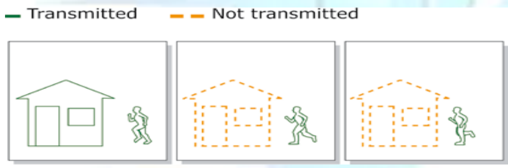


Fig 5. H.264 Coding Sequence

The figure 5 shows that with difference coding (used in most video compression standards including H.264), only the first image (I-frame) is coded in its entirety. In the two following images (P-frames), references are made to the first picture for the static elements, i.e. the house, and only the moving parts, i.e. the running man, is coded using motion vectors, thus reducing the amount of information that is sent and stored.

## III.MOTION ESTIMATION

One of the main devices, needing a lot of computation, is the motion estimator whose role is to find in a limited area of the previous image centered around the reference block position and called search window, the position of the block most similar to the reference block: this position is called the motion vector . In most cases, the picture is defined by the three components Y (luminance) U and V (chrominance), but the motion estimation is only made on the luminance component, the same resulting motion vector being applied to all components. The comparison of the reference block with each possible block of the search window needs a lot of computation. An expression for the distance between the reference block and a predicted block is:

$$D(di,dj) = \min [Ref(m,n) - Pred(m+di,n+dj)]$$

Where: Ref (m,n) = reference block pixel

Pred(m+di,n+dj) = predictor pixel with displacement di in vertical and dj in horizontal directions. Computing a distance between two blocks 8x8 for instance needs 64 subtractions, 64 multiplications and 64 accumulations. For a -8 to +7 possible displacement in both directions (horizontal and vertical), that is 256 possible predicted blocks, and implies 256 x 64 x3 = 49152 operations to do during the 64 cycles of the reference block before being able to extract a motion vector.

.

The MSE is calculated as follows:

$$MSE(x, y, r, s) = \frac{1}{256} \sum_{i=0}^{15} \sum_{i=0}^{15} ( A_{(x+i,y+j)} - B_{((x+r)+ix(y+x)+j)} )^2$$

Due to the square operation on the differences, this operation is less commonly used. Instead, the MAD is used more often and it is calculated as follows:

$$MAD(x, y, r, s) = \frac{1}{256} \sum_{i=0}^{15} \sum_{i=0}^{15} (A - B_{((x+r) + ix(y+x)+j)})^2$$

The vector (x; y) denotes the location of the to be encoded macro block in the current frame. Due to the computational simplicity of the MAD, it is being used more often than the MSE. The MAD can be rewritten to:

$$MAD(x,y,r,s)=SAD(x,y,r,s)/256$$

The division by 256 in (binary) computer arithmetic is translated into an easy shifting the final SAD result by 8 bits. Therefore, we are focusing solely on the SAD in the remainder of this paper. All the absolute operations of the SAD operation can be performed serially, per column in parallel, per row in parallel, or all 256 operations in parallel. While it is possible to perform all the operations serially, this approach is time-consuming and not efficient performance-wise .

Considering that the completely parallel approach is a simple extension and this alleviates the need for special reordering hardwareThe complete SAD operation can be rewritten to:

$$SAD(x,y,r,s) = \sum_{j=0}^{15} SAD16(x,y,r,s)$$

With the SAD 16j being defined as:

$$SAD\ 16j(x,y,r,s)=\sum_{i=0}^{15} | A_{(x+i,y+j)} - B_{((x+r)+iz(y+x)+j)} |$$

ISSN: 2320 - 8791
www.ijreat.org

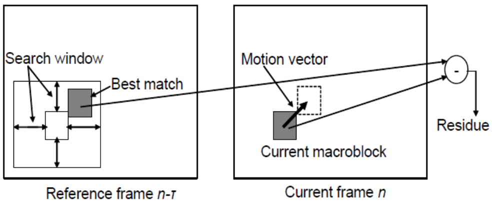


Fig 6 Motion Estimation

*A.Motion vector*

Express the relative horizontal and vertical offsets(mv1,mv2), or motion, of a given block from one frame to another.Each block has its own motion vector.

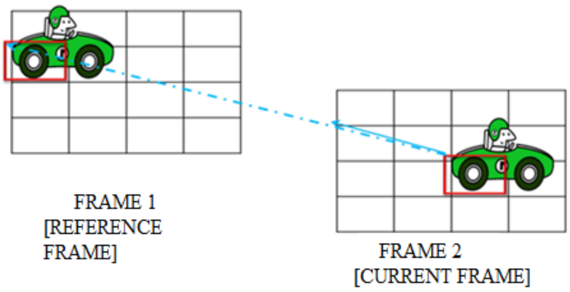


Fig 7. Motion vector

*B.Motion vector field*

Collection of motion vectors for all the blocks in a frame. Advantages are, Good, robust performance for compression. Resulting motion vector field is easy to represent (one MV per block) and useful for compression. Simple, periodic structure, easy VLSI implementations

*C.Macroblock*

Macroblock is an image compression component and technique based on discrete cosine transform used on still images and video frames. Macroblocks are usually composed of two or more blocks of pixels. The size of a block depends on the codec and is usually a multiple of 4. In MPEG2 and other early codecs, a block had a fixed size of 8×8 pixels; a macroblock comprised four Y blocks, one Cb block and one Cr block, defining a 16×16 pixel square of YCbCr 4:2:0. In more modern codecs such as h.263 and h.264 the overarching macroblock size is fixed at 16×16 pixels.

*D.HFPS ALGORITHM*

Motion estimation process can be divided into two steps: the integer pixel ME and fractional pixel ME. The search range of fractional pel ME in H.264/AVC JVT reference software is fixed to + 3 for quarter pel accuracy case so in many cases people prefer to choose FS at this stage for simplicity.

The conventional Hierarchical Fractional Pixel Search (HFPS) algorithm that has been adopted in the reference software at ¼ pel accuracy needs to check 17 search points. Hence reducing the computational load for fractional pel motion search is both necessary and significant. Figure 1 shows the conventional hierarchical fractional pel search (HFPS) method in H.264. First it examines eight ½ pel positions surrounding the best integer pixel position and obtains the best ½ pel MV. Then it checks eight ¼ pel positions to obtain the best ¼ pel MV.

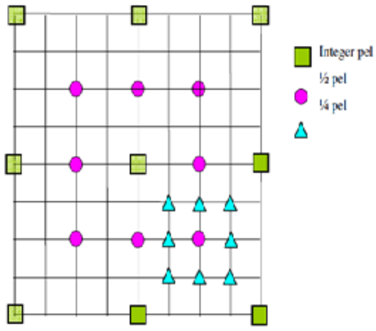


Fig 8. Hierarchical Fractional Pixel Search(HFPS)

## IV.IMPLEMENTATION OF MBFPS ALGORITHM

Hybrid coding framework is first successfully adopted in H. 261 and after that various moving picture coding standards, such as MPEG-1, MPEG-2, H.263, MPEG-4 and JVT are all based on this framework. Motion estimation is the most important part inside hybrid coding framework in exploiting the high temporal redundancy between successive frames to improve video coding efficiency. And motion estimation is also the most time consuming part in the hybrid coding framework.

Generally motion estimation is conducted into two steps: first is integer pel motion estimation; and the second is fractional pel motion estimation around the position obtained by the integer pel motion estimation (we name it the best integer pel position). For fractional pel motion estimation, 1/2-pel accuracy is frequently used (H.263, MPEG-1, MPEG-2, MPEG-4), higher resolution motion vector are adopted recently in MPEG-4 (1/4-pel accuracy) and JVT (1/4, 1/8-pel accuracy) to achieve more accurate motion description and higher compression efficiency

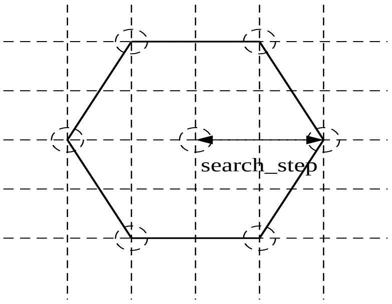


Fig 9.Hexagon grid search pattern

Hexagon search pattern is used in our fast integer pel motion estimation algorithm, the six search points around the center can be described as a set:

$$\Omega(search\_step, center\_x, center\_y)$$

*A.Fast fractional pel Motion Estimation*

We proposed a Medium Biased Fractional Pel Search (MBFPS) algorithm for fast fractional pel motion estimation. Using formula (1), pred_mv here is defined as the fractional pel unit, so it includes the information of predicted integer pel motion vector and predicted fractional pel motion vector. Therefore we could extract the predicted fractional pel motion vector by using this formula:

$$frac\_pred\_mv=(pred\_mv-mv)\%\beta$$

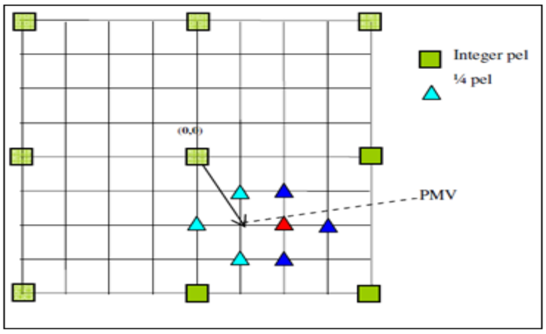


Fig 10. Medium biased fractional pixel

V.RESULTS

This Fig 11. shows the simulation result for Motion Estimation

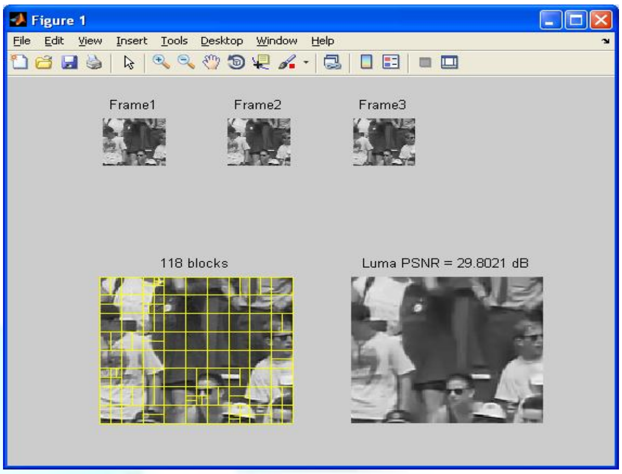


Fig 11. Motion estimation

Table shows the comparison between the HFPS and MFPS

| METHODS | INTERPEL | ½ PEL | ¼ PEL | SAD |
|---|---|---|---|---|
| HFPS | 9 | 8 | 8 | 27 |
| MFPS | 9 | 0 | 6 | 15 |

## V.CONCLUSION AND FUTURE WORK

This thesis presents a Median Biased Fractional Pel Search method for H.264/AVC, including accurate initial prediction, dynamic search range, simplified search flow, adaptive search pattern, intelligent early termination, and fast sum-absolute-transformed-differences table lookup. Integrate our method into the latest advanced video coding scheme H.264/AVC. Experimental results show that the proposed method offers an average of 55% complexity reduction with competitive rate-distortion performance when compared with those of the state-of-the-art methods. In addition, our proposed method offers smoother motion vectors which reflect more true motion vectors.

A few areas closely related to motion estimation are worth to be explored. First, modifying the current motion estimation methods to generate true motion vector would benefit applications such as video surveillance and moving object detection. Adaptive error-resilient motion estimation with optimal complexity-rate-distortion is vital for future wireless low power video applications. A new framework and theoretical analysis by adding one more dimension, i.e. error-resilience over complexity-rate-distortion three dimension paradigms is even challenging.

## REFERENCES

[1] Jose L. Nunez-Yanez, Atukem Nabina, Eddie Hung, and George Vafiadis , " Cogeneration of Fast Motion Estimation Processors and Algorithms for Advanced Vedio Coding",IEEE Transactions On Very Large Scale Integration(VLSI) System, vol 20,No.3,March 2012

[2] J. Nunez-Yanez, E. Hung, and V. Chouliaras, "A configurable and programmable motion estimation processor for the H.264 video codec,"in Proc. Int. Conf. Field Programmable Logic Applic., Sep. 2008, pp.149–154.

[3] G. J. Sullivan, T. Wiegand, and P. Corporation, "Rate-distortion optimization for video compression" IEEE Signal Process. Mag., vol. 15,pp. 74–90, 1998.

[4] S.-C. Cheng and H.-M. Hang, "A comparison of block-matching algorithms mapped to systolic-array implementation," IEEE Trans. Circuits Syst. Video Technol., vol. 7, no. 5, pp. 741–757, Oct. 1997.

[5] T. Dias, S. Momcilovic, N. Roma, and L. Sousa, "Adaptive motion estimation processor for autonomous video devices" EURASIP J. Embedded Syst., vol. 2007, no. 1, p. 41-41, 2007.

[6] K. Babionitakis, G. Doumenis, G. Georgakarakos, G. Lentaris, K. Nakos, D. Reisis, I. Sifnaios, and N. Vlassopoulos, "A real-time motion estimation FPGA architecture," J.Real-Time Image Process., vol. 3, no. 1–2, pp. 3–20, Mar. 2008.