RESEARCH ARTICLE OPEN ACCESS

A New Cross Diamond Search Motion Estimation Algorithm for HEVC

A. Karthikeyan, Dr. Amos H jeeva oli
(K.C.G College of technology)
(K.C.G College of technology)

ABSTRACT
In this project, a novel approach for motion estimation is proposed. There are few block matching algorithm existing for motion estimation. In motion estimation a new cross diamond search algorithm is implemented compared to diamond search it uses less search point. Because of this we can reduce the computational complexity. The performance of the algorithm is compared with other algorithm by means of search points. This algorithm achieves close performance than that of three step search and diamond search. Compared to all the algorithm cross diamond uses less logic elements, delay and power dissipation.
KEYWORDS: Zig-Zagscan, Cross diamond search, Diamond search, Delay.

I. INTRODUCTION:

The increasing demand to incorporate video data into telecommunications services, the corporate environment, the entertainment industry, and even at home has made digital video technology a necessity. A problem, however, is that still image and digital video data rates are very large, typically in the range of 150Mbits/sec. Data rates of this magnitude would consume a lot of the bandwidth, storage and computing resources in the typical personal computer. For this reason, Video Compression standards have been developed to eliminate picture redundancy, allowing video information to be transmitted and stored in a compact and efficient manner.

II. VARIOUS VIDEO COMPRESSION STANDARDS

MPEG-1:

MPEG-1, the first lossy compression scheme developed by the MPEG committee, is now a days used for CD-ROM video compression and as part of previous Windows Media players. TheMPEG-1 algorithm uses Discrete Cosine Transform (DCT) algorithm which initially convert each image into the frequency domain and then process these in frequency coefficients to optimally reduce a video stream to the required bandwidth.

MPEG-2:

The MPEG-2 compression standard is used to overcome the requirements of compressing higher quality videos. It is the combination of lossy video compression and lossy audio data compression methods, which is used to store and transmit pictures using currently available storage media and transmission bandwidth.

MPEG-3:

The MPEG committee originally intended that an MPEG-3 standard would evolve to support HDTV, but it turned out that this could be done with minor changes to MPEG-2. So MPEG-3 never happened, and now there are profiles of MPEG-2 that support High Definition Television (HDTV) as well as Standard Definition Television (SDTV).

MPEG-4:

MPEG-4 is a method of defining compression of audio and visual (AV) digital data. It is used in the compression AV (audio and visual) data for web and CD distribution voice and broadcast television application. The main aim of the MPEG-4 standard is to find solution for two video transport problems: sending video over low bandwidth channels such as the video cell phones and internet, and achieving better compression compared to MPEG-2 for broadcast signals.

H.264/AVC:

As MPEG-4 fails to improve compression performance for full broadcast signals, the H.264 is developed to achieve a improvement of 2:1 over MPEG-2 on full-quality SDTV and HDTV.H.264/MPEG-4 AVC is a block oriented motion compensation based codec standard jointly developed by the ISO/IEC Moving Picture Expert Group (MPEG) and ITU-T Video Coding Expert Group (VCEG). The project partnership effort is known as the Joint Video Team (JVT). It is also known as MPEG-4 part 10 AVC (Advanced Video Compression).

III. EXISTING SYSTEM:

The video resolution required for many types ofvideo content has increased as technology has advanced. For the real-time encoding of the high resolutions such as full high definition (FHD), quad-FHD (QFHD) and beyond, various fast motion estimation (ME) algorithms have been researched. Caches are used for many fast MEs in a hardware-based encoder, in order to increase local memory utilization and thereby reduce external memory access. In a multi-core environment for high resolution videos, access conflicts directly affect the computation time. In this paper, various types of caches are compared in terms of the size, hit ratio, cache port conflicts and hardware overhead. To reduce the amount of cache access associated with the basic shared cache, zigzag snake scan and selective data-storage schemes are proposed for integer and fractional MEs, respectively. Additionally, the cache access arbitration hides the computation delay which arises due to a cache port conflict in a pipeline system. The proposed schemes are applicable for the existing cache design achieving a good scalability in a multi-core environment. Simulation results show that the ME computation time reduced by the proposed schemes is comparable to that of the dual-port shared cache which shows the least amount of port conflicts.

IV. CONVENTIONAL SNAKE CHAIN:

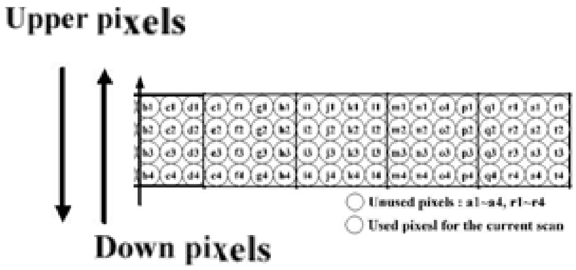

Figure 1.1

Figure 1.1 shows a snapshot of a search in the conventional snake scan. Five squares represent 4×4 cache blocks, whereas each circle represents a pixel. The arrows denote the scan order. To search the points from b4 to b1 pixels, the reference pixels from b1 to q4 are required; thus, five 4×4 blocks (from a1 to t4) are loaded from the cache to the internal buffer. The pixels from a1 to a4 and from r1 to t4 are not used. To keep searching the upper points of column b after searching from b4 to b1, the internal buffer is filled with the newly loaded cache blocks. Next, to search the points from c1 to c4, the IME engine should read the five 4×4 blocks from a1 to t4 again. This repeated loading of the same cache blocks also occurs when searching the d4 to d1 pixels and the e1 to e4 pixels. As a result, precisely the same cache blocks should be loaded into the internal buffer four times to execute the IME operation for 16 search points. These repeated cache accesses to read the same reference data cause port conflicts. Unlike the pixel-based internal buffer, in the block-based cache design, the conventional snake scan shows low internal buffer utilization because the cache access data in the internal buffer cannot be fully reused.

V. PROPOSED SYSTEM:

In the proposed system we are using zig-zag snake chain instead of snake chain, so we can reduce the delay.

VI. ZIG ZAG SNAKE CHAIN:

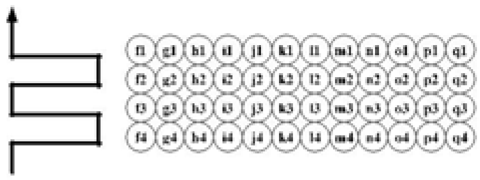

Figure 1.2Zigzag snake chain

Figure 1.2 shows the proposed zigzag snake scan order. After searching b4, the arrow moves to c4→d4→e4 and not to b3→b2→b1. Reference pixels r4, s4 and t4, which are already loaded, are used for searching the c4, d4 and e4 points. After searching point e4, the scan direction moves to point e3 and then moves from the right to the left pixels. To support the horizontal shift, which is the main difference between the snake scan and the proposed zigzag scan, registers 19×16 in size for the reference pixels are used instead of those 16×16 in size. Thus, 19×15 reference pixels are reused and 19×1 pixels are provided from the internal buffer to the IME engine at one time. Although there is a small amount of hardware overhead associated with registers 3×16 in size, the zigzag scan decreases the frequency of cache accesses significantly. By fully reusing the cache blocks in the internal buffer, IME computes 16 search points from b1 to e4 with only one cache access. In this example, the number of cache accesses decrease by four times compared to the conventional snake scan order. This reduction in the number of cache accesses directly leads to a decrease in the number of cache port conflicts. In this paper we are implementing this Zig-Zag scan to all the three algorithms

VII. BLOCK MOTION MATCHING:

Block-based motion estimation is one of the major components in video compression algorithms and standard. The objective of motion estimation is to reduce temporal redundancy between frames in a video sequence and thus achieve better compression. In block-based motion estimation, each frame is divided into a group of equally sized macro blocks and to find the best matching macro blockin the reference frame to the macro block being encoded in the current frame. Once the best match is located, only the difference between the two macro blocks and motion vector information are compressed.

The most commonly used matching criterion is the sum of absolute differences (SAD), which is chosen for its simplicity and ease of hardware implementation. For an *M* x *N* block, where Sl(*x*,*y)* is the pixel value of frame *l* at relative position x,*y*from the macro block origin and Vi = (*dx*, *dy)* is the displacement vector, SAD can be computed as follows

$$SAD(V_i)=\sum_{x=0}^{M}\sum_{y=0}^{N}|S_l(x,y)-S_{l-1}(x+dx,y+dy)|$$

VIII. RESULT AND ANALYSIS :

SPEED (FMAX SUMMARY):

Timing analysis is a process of analyzing delays in a logic circuit to determine the conditions under which the circuit operates reliably. These conditions include, but are not limited to, the maximum clock frequency (f-max) for which the circuit will produce a correct output. Computing f-max is a basic function of a timing analyzer. The timing analyzer can be used to guide Computer-Aided Design tools in the implementation of logic circuits. For example, the circuit in Figure 1 shows an implementation of a 4-input function using 2-input AND gates. Without any timing requirements, the presented solution is acceptable. However, if a user requires the circuit to operate at a clock frequency of 250 MHz, the above solution is inadequate. By placing timing constraints on the maximum clock frequency, it is possible to direct the CAD tools to seek an implementation that meets those constraints.

IX. POWER DISSIPATION:

Device families have different power characteristics. Many parameters affect the device family power consumption, including choice of process technology, supply voltage, electrical design, and device architecture. Power consumption also varies in a single device family. A larger device consumes more static power than a smaller device in the same family because of its larger transistor count. Dynamic power can also increase with device size in devices that employ global routing architectures. The choice of device package also affects the ability of the device to dissipate heat. This choice can impact your required cooling solution choice to comply to junction temperature constraints. Process variation can affect power consumption. Process variation primarily impacts static power because sub-threshold leakage current varies exponentially with changes in transistor threshold voltage. So, you must consult device specifications for static power and not rely on empirical observation. Process variation has a weak effect on dynamic power.

X. OVER ALL COMPARISION:

	CDS	DS	TSS
AREA	378	383	392
POWER	90.47mw	139.40mw	101.46mw
DELAY	214.55mhz	205.34mhz	473.04mhz

XI. CONCLUSION:

Here we implemented a scalable Motion estimation Search algorithm for computationally efficient block motion estimation for image compression based motion speed. These techniques can be applied for both spatial and temporal image. Because of the compact shape of the search pattern and step size it outperforms other existing algorithms such as TSS, NTSS, and DS in terms of computational efficiency with a better performance. This algorithm can be used in video coding standards such as MPEG-4, H.264 AVC because of its ease of implementation, better performance with reduced computational complexity and high speed searching.

XII. FUTURE WORK:

Future work will concentrate on reducing delay and power consumption in motion estimation by working on BRAM concept and factor sharing. A special emphasis will also be laid on a reconfigurable architecture for the same to reduce the area.

REFERENCES

[1.] K. B. Kim, Y. Jeon, and M.-C. Hong, "*Variable Step Search Fast Motion Estimation for H. 264/AVC Video Coder*," IEEE Trans. Consumer Electron., vol. 54, no. 3, pp. 1281-1286, Aug. 2008.

[2.] Ishfaq Ahmad, Senior Member, IEEE, Weiguo Zheng, Member, IEEE, Jiancong Luo, Member, IEEE, and Ming Liou, Life Fellow, IEEE "*A Fast Adaptive Motion Estimation Algorithm*" IEEE transactions on circuits and systems for video technology, vol. 16, no. 3, march 2006.

[3.] Chi wai lam Lai man po and Chum ho cheung, "*A new cross-diamond search algorithm for fast block matching motion estimation*" Department of Electronic Engineering City University of Hong Kong, Hong Kong SAR.

[4.] 4) Yi luo, "*Fast Adaptive Block Based Motion Estimation for Video Compression*", Department of Electrical Engineering and Computer Science and the Russ College of Engineering.

[5.] Jong namkim Tae sun choi, "*A fast three step search algorithm with minimum checking points using unimodal error surface assumption*" IEEE Transaction vol.44 pp. 638-648.
[6.] Cezhu, Xiao lin, Lap puichau, Keng pang lim, Hock annang, Choo yin ong, "*A novel hexagonal based search algorithm for fast block motion estimation*, centre for signal processing, school of electrical & electronic engineering.
[7.] T. C. Chen, S. Y. Chien, Y. W. Huang, C. H. Tsai, C. Y. Chen, T. W. Chen, and L.-G. Chen, "*Analysis and architecture design of an HDTV720p 30 frames/s H.264/AVC encoder*," IEEE Trans. Circuit Syst. Video Technol., vol. 16, no. 6, pp. 673-688, June 2006.
[8.] Y. K. Lin, C. C. Lin, T. Y. Kuo, and T. S. Chang, "*A Hardware-Efficient H.264/AVC Motion-Estimation Design for High-Definition Video*," IEEETrans. Circuits Syst. I, vol. 55, no. 6, pp. 1526-1535, July 2008.
[9.] Li Zhang and W. Gao, "*Reusable Architecture and Complexity-Controllable Algorithm for the Integer/Fractional Motion Estimation of H.264*," IEEE Trans. Consumer Electron., vol. 53, no. 2, pp. 749-756, May 2007.