INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

STUDY OF HASH BASED BLOCK MATCHING FOR MOTION ESTIMATION

KU. DIPTI V. JILHARE, DR. S. Y AMDANI

Computer Department, BNCOE, Pusad, INDIA

Accepted Date: 15/03/2016; Published Date: 01/05/2016

Abstract: With the increasing popularity of technologies such as Internet streaming video and video conferencing, video compression has became an essential component of broadcast and entertainment media. Motion Estimation (ME) and compensation techniques, which can eliminate temporal redundancy between adjacent frames effectively, have been widely applied to popular video compression coding standards such as MPEG-2, MPEG-4. A motion estimation algorithm is to improve the performance of the existing searching algorithms at a relative low computational cost. In this paper will be provide survey of fastest block matching algorithm in motion estimation. Concentrate on study of "Hash based block matching in motion estimation. We propose a hash-based block matching scheme for the motion estimation process, which enables frame level block searching. In the proposed scheme, the blocks sharing the same hash values with the current block are selected as prediction candidates. Then the hash-based block selection is employed to select the best candidates. Achieve best coding efficiency. The proposed scheme, bit rate saving and time reduction.

Keywords: Motion estimation, Hash based block matching, Motion compensation, HEVC.

Corresponding Author: KU. DIPTI V. JILHARE

Access Online On:

www.ijpret.com

How to Cite This Article:

Ku. Dipti V. Jilhare, IJPRET, 2016; Volume 4(9): 1448-1458

PAPER-QR CODE

INTRODUCTION

Due to the limitations in the available bandwidth and storage space for high quality multimedia content like - video broadcasting and DVD video data, video compression has become very much necessary to keep up the ever growing demand, by maintaining the quality in decoded video. The purpose of the motion estimation (ME) and compensation is reduction of redundancy caused by interframe correlation of movement objects. However, the estimation and coding of movement vectors should be appropriated to computational costs and bit rates at the perspective high compression systems. That's why relationship between accuracy of movement estimation and simplicity of the description vector fields is very important. Better motion estimation means higher space decorrelation of prediction errors in time area.

In this paper, we present a hash-based block matching scheme to enable large range block search in motion estimation of HEVC. In the proposed scheme, the hash value of every block is obtained using a carefully designed hash function .Then the blocks sharing the same hash values with the current coding block will be selected as candidates. Finally, hash-based block selection is designed to find the best matching blocks for the current block

There are four advantages in the proposed scheme. First, the coding efficiency of HEVC range extension on screen contents is significantly improved by enabling full-frame level block searching. Second, by using the hash-based block matching, the proposed scheme will not bring great computation burden. Third, RDO based block matching proposes a more efficient block matching solution which is suitable for devices with high performance. Fourth, the proposed scheme doesn't change the syntax of HEVC range extension, which makes it compatible with existing standards.

2. BLOCK MATCHING ALGORITHM

The block-matching algorithm (BMA) [1] is the most popular and widely used algorithm for motion estimation due to its simplicity and reasonable performance. In BMA[1], an image frame is divided into non-overlapping rectangular blocks with equal or variable block sizes. The pixels in each block are assumed to have the same motion. The motion vector (MV) of a block is estimated by searching for its best match within a search window in the previous frame. The distortion between the current block and each searching block is employed as a matching criterion. The resulting MV is used to generate a motion compensated prediction block. The motion-compensated prediction difference blocks (called residue blocks) and the MVs are encoded and then sent to the decoder[1].

3. FULL SEARCH METHOD

Many ME algorithms already exist, among all the estimation algorithms, full-search (FS) block-matching algorithm is a popular ME algorithm. The concept of block matching algorithm to find the movement of each block between the previous and current frame called displacement vectors (MVs) is depicted in Figure ***.1***

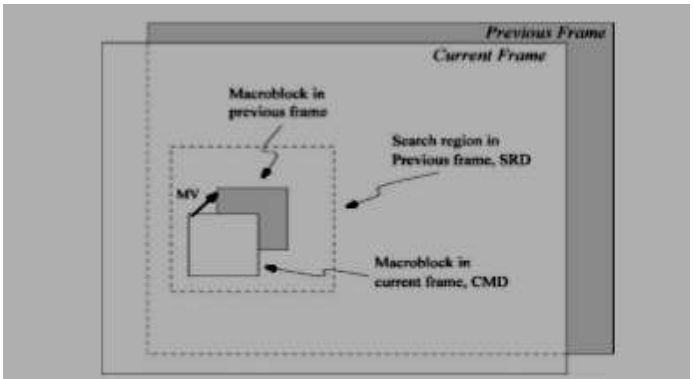

Figure 1. Block Matching Motion Estimation

In Full search algorithm motion vector is calculated in two stages, namely the calculation of the SAD for each displacement vector, followed by methods for finding the smallest SAD values[2]. In general, the common feature in all fast BMAs is the trade-off between quality and search speed. Increasing speed as much as possible, while preserving quality, is the major goal of all fast BMAs.

Full search is still attractive for the high end applications, at the cost of increased computation and power consumption because of its performance in terms of PSNR. Many hardware architectures are proposed for full search motion estimation. These architectures try to compute the SAD at all, such locations in the search window. Popular hardware architectures are the partial propogate SAD architecture. Hardware architectures access the blocks in a raster scan to maximize pixel reuse. The main disadvantage of these architectures is the inability to start the search around a search center of the search window of a reference frame and early termination techniques. The full search (FS)[2] is global optimal and most straightforward algorithm because it searches the entire search window for the best matching block. However, its only drawback is a heavy computational load.

4. STUDY ON HASH-BASED BLOCK MATCHING IN MOTION ESTIMATION

To address the above issues, we have proposed modified Hash based block matching scheme in motion estimation with search step is frame level block searching compared to the traditional search. This results in skipping of a certain number of blocks. To further speed up, a trace and off diagonal sum of the current frame block and the reference frame block is matched to find the nearly matching block.

The basic idea of the hash-based block matching scheme is to find the prediction block for the current block by comparing the hash values of the current block with the hash values of the blocks in the reconstructed regions or reference frames. The first-level and second-level hash values of the current block are first generated[3]. Then the hash values of the blocks in the reconstructed regions are compared with those of the current block. The first-level hash value is first used to find the candidates approximated to the current block. If more than one block is found, the best prediction block is chosen from the candidates by comparing the second-level hash values. The current block is then encoded and reconstructed using the selected block as prediction. After that, there constructed block is updated to the reconstruction buffer. Then all newly available blocks in the reconstruction regions are hashed and updated to the hash dictionary. The hash dictionary constructed during the encoding process of the current frame will be used for inter motion estimation[3].

So it needs variety combination of techniques, such as motion starting point, motion search patterns and adaptive search control to terminate the search and many more that makes ME a robust. Hence there is a requirement for improved algorithms and hardware architectures which are suitable for real time applications. To find the state-of-the-art method, a literature survey is carried out[3].

5. LITERATURE SURVEY:

In past, many techniques have been developed to accelerate the block matching process. We classify these techniques into three categories. The techniques in the first category save the computations by reducing the number of the positions searched. Therefore, the obtained minimum of the matching error may only be a local minimum within the search set, S. The techniques in the second category, on the other hand, try to reduce the computational cost of the matching error for each search position. Whether this kind of techniques can obtain the global minimum or not depends on how we compute and compare the matching errors. The

techniques in the first and second categories can be combined to further improve the efficiency and this kind of hybrid methods are classified as the third category [4].

Block based Motion Estimation has been adopted [2-5] to reduce the temporal redundancy between frames.

Full search involves the computation of SAD at each location in the search window. For a search window of size +/- P pixels, the number of search locations is (2P+1)2. For a search window of 32x32 and a block size of 16x16, a total of 289 locations is searched to find the best match with the minimum SAD value. This results in significant computational complexity.

Many algorithms have been proposed to reduce the computational complexity of full search motion estimation. Some of the popular ones are the Three Step Search (TSS [5]), the New Three Step Search (NTSS [6]), the Four Step Search (4SS [7]), the Diamond Search (DS [8]) and the Adaptive Rood Pattern Search (ARPS [9-10]). These algorithms try to do Small Square (TSS, NTSS, 4SS) or diamond shaped (DS) search around a search center, and refine the search around the best matching block. Early termination techniques based on the SAD threshold values are used to reduce the computation cost. Algorithms like ARPS employ sophisticated search center prediction as the start point. Though these algorithms address computational cost well, but the performance in terms of PSNR is close to FS algorithm[10-13].

Concerning the VLSI implementation, most of these fast algorithms, e.g., the three-step search (TSS) [5], have the drawbacks of irregular data flow and high control overhead, while the full-search BMA has the advantages of regular data flow and low control overhead [11]. Recently, a number of algorithms with regard to the pattern matching problems[12-15] make use of integral projections to simplify the computational complexity of the pattern matching operation. However, all of the previous research work on motion estimation using integral projections has never provided any optimality preserving ability like the FBMA. Integral projections are good features describing the block mean intensity and the edge location and orientation in a block of pixels, and are most likely to be different for different blocks[11-17].

In this letter, a fast full-search BMA (FFBMA), which is also based on the uses of integral projections, is presented to provide much faster motion estimation than that using the traditional FBMA, while preserving the optimality of estimate accuracy[11-15].

6. LONG RANGE MOTIONS IN INTER FRAMES

Motion estimation and motion compensation are main techniques to remove the redundancy of inter frames. In HEVC and its range extension, the default search range of motion estimtion is (64*2+w)*(64*2+h), where w and h are the width and height of the current prediction unit respectively. In sequences with natural contents, even when the object is moving fast, the motion range between successive frames is relatively small. However, long range motions as large as half a frame of ten happen in the scenarios like we be browsing and document editing. As shown in Fig. 2, the texts (boxed by red rectangles) move up 128 pixels between two successive frames. The matched blocks will not be found using the default search range[18-21].

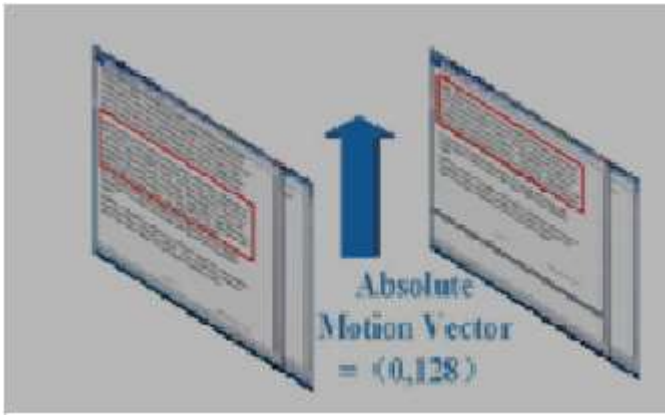

Fig.2. long range motion from two frame 1

7. HASH-BASED BLOCK MATCHING SCHEME

A. FRAMEWORK OF HASH-BASED BLOCK MATCHING

The basic idea of the hash-based block matching scheme is to find the prediction block for the current block by comparing the hash values of the current block with the hash values of the blocks in the reconstructed regions or reference frames. As shown in Fig. 3, The first-level and second-level hash values of the current block are first generated. Then the hash values of the blocks in the reconstructed regions are compared with those of the current block. The first-level hash value is first used to find the candidates approximated to the current block.

If more than one block is found, the best prediction block is chosen from the candidates by comparing the second-level hash values. The current block is then encoded and reconstructed using the selected block as prediction. After that, there constructed block is updated to the reconstruction buffer. Then all newly available blocks in the reconstruction regions are hashed

and updated to the hash dictionary. The hash dictionary constructed during the encoding process of the current frame will be used for inter motion estimation [19-20].

B. HASH-BASED BLOCK SELECTION

Let be the hashed blocks in the reconstructed regions of the current frame or the reference frames, where is the block and is the number of hashed blocks. Let be all possible first-level hash values. The hashed blocks are organized using a hash dictionary based on their first-level and second-level hash values.

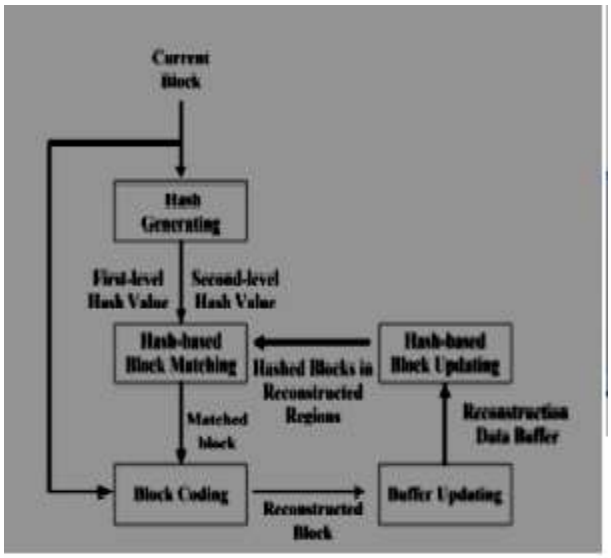

Fig 3. Framework of the proposed system

The blocks with the same first-level hash value will be stored to the same position of the hash dictionary. The blocks with the same first-level hash values and different second hash values will be stored as a list. Thus the blocks similar to the current block can be found in time 0(1). As shown in Fig. 4, given an input block with hash values (L3, 234), the position of the blocks approximated to the current block can be determined by the first-level hash value . The selected blocks (B31,B32,B33) are further checked by comparing these second-level hash values. The block with the matched second-level hash value (B33) is selected as the best prediction block. After the current block is encoded, the hash-based block updating is employed to filter out identical blocks from all newly available candidates which need to be updated to the hash dictionary. Only the blocks with first-level and second-level hash values different from those in the hash dictionary will be included in the hash dictionary[3-20].

C. HASH FUNCTION DESIGN

Two hash functions are adopted in the proposed scheme to find the best prediction block. The first hash function used in the hash-based block matching is designed to find the prediction blocks approximated to the target block. However traditional hash functions like CRC [20] can only be used to find blocks identical to the input block. As a result, traditional hash functions do not help to find blocks which approximates to the input block.

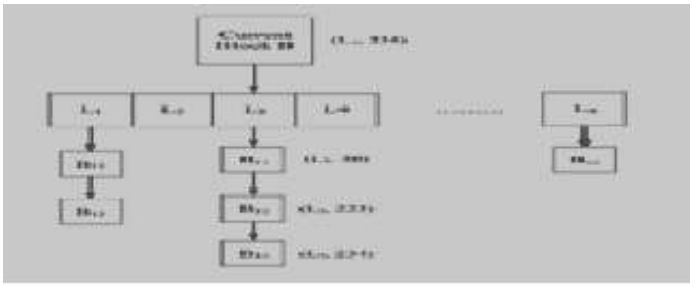

Fig 4. Hash based block matching

The proposed hash function for the first-level hash values consists of three 8-bit components , μ,r and c . The first 8-bit μ is generated from the average value of the target block. The second and third 8-bit components are generated by performing directional XOR operations on the target block. The second and third 8-bit components convey the structure information of the target block. Let [i , j] be the8*8 block with 8-bit pixel values, the first 8-bit μ is calculated as

$$\mu - \frac{\sum_{0\le i<8,0\le j<8} B[i,j]}{64}. \quad (1)$$

For the second and third 8-bit, a binary level map M is first derived using the average value μ of the target block

$$M[i,j]=\begin{cases}1, & \text{when } B[i,j]>\mu\\ 0, & \text{otherwise}\end{cases}, 0\le i<8;\ 0\le j<8. \quad (2)$$

The second 8-bit r is generated by performing XOR operations along rows of the level map M. The eight bits of r is deduced as follows:

$$r[j] = M[0,j]\oplus M[1,j]\oplus M[2,j]\oplus M[3,j]\oplus M[4,j] \oplus M[5,j]\oplus M[6,j]\oplus M[7,j] \quad 0\le j<8 \quad (3)$$

where $\oplus$ denotes the XOR operation. The third 8-bit c is generated by performing XOR operations along columns of the level map M. The eight bits of c is deduced as follows:

$$c[i] = M[i,0]\oplus M[i,1]\oplus M[i,2]\oplus M[i,3]\oplus M[i,4] \oplus M[i,5]\oplus M[i,6]\oplus M[i,7] \quad 0\le i<8. \quad (4)$$

Finally the three components μ,r,c are combined together to generate the hash value which can be calculated as follows

h=(μ<<16)+(r<<8)+c

where << denotes the left shift operator .

The second hash function is used to locate the identical blocks. The CRC[20] with 8bits, which is very suitable for filtering out identical blocks from candidates, is used as the second-level hash function in the proposed scheme[22].

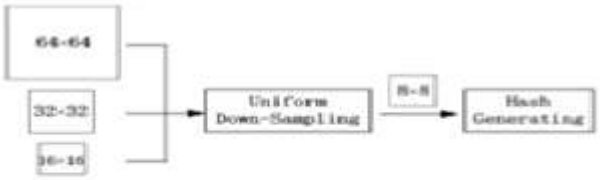

Fig 5. Uniform down sampling process

8. CONCLUSION

Screen content has emerged as an important category of video source. High compression efficiency of screen on extent is highly desired. Among the all motion estimation methodologies, the block matching received very much attention by researcher because of their simplicity and regularity. In this paper, motion estimation, criteria for movement estimation performance. Then propose a hash-based block matching scheme for the motion estimation process, which enables frame level block searching. In the proposed scheme, the blocks sharing the same hash values with the current block are selected as candidates. Hash-based block selection is employed to select the best candidates to achieve the best coding efficiency.

9. REFERENCS

1. Chung-Ming Kuo, Nai-Chung Yang, I-Chang Jou1 and Chaur-Heh Hsieh, “A Novel Prediction-Based Asymmetric Fast Search Algorithm for Video Compression”, Ming Chung University Gui-Shan, 333, Taoyuan, Taiwan, R.O.C.
2. Manu T. M, Linganagoud Kulkarni and Basavaraj S. Anami “A Fast Trace Based Spiral Search Architecture for Motion Estimation and its Implementation Using FPGA” International Journal of Communication Technology for Social Networking Services Vol.3, No.1 (2015), pp.23-32.

3. Weijia Zhu, Wenpeng Ding, Jizheng Xu, Yunhui Shi, and Baocai Yin "Hash-Based Block Matching for Screen Content Coding" IEEE TRANSACTIONS ON MULTIMEDIA, VOL.17, NO.7, JULY 2015 935.

4. Yong-Sheng Chen, Yi-Ping Hung, and Chiou-Shann Fuh "Fast Block Matching Algorithm Based on the Winner-Update Strategy" IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 10, NO. 8, AUGUST 2001 http://dx.doi.org/10.14257/ijctsns.2015.3.1.03 ISSN: IJCTSNS Copyright ⓒ 2015 SERSC.

5. I. Ahmad, W. Zheng, J. Luo, and M. Liou, "A fast adaptive motion estimation algorithm," IEEE Trans. on Circuits and Systems for Video Technology, vol. 16, (2006) March, pp. 4280-438.

6. W. I. Chong, B. Jeon, and J. Jeong, "Fast motion estimation with modified diamond search for variable motion block sizes," inProc. Int.Conf. Image Process. , vol. 3, (2003) September, pp. 371–374.

7. R. Li, B. Zeng, and M. L. Liou, "A new three-step search algorithm for block motion estimation," IEEE Trans. Circuits Syst. Video Technology, vol. 4, no. 4, (1994) August, pp. 438–442.

8. 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, no. 3, June, pp. 313–317.

9. A. W. Zheng, J. Luo, and M. Liou, "A fast adaptive motion estimation algorithm," IEEE Trans. Circuits Syst. Video Technol., vol. 16, no. 3, (2006) March, pp. 420–438.

10. S. Goel, Y. Ismail, and M. A. Bayoumi, "Adaptive search window size algorithm for fast motion estimation in H.264/AVC standard," in Proc. Midwest Symp. Circuits Syst., (2005) August, pp. 1557–1560.

11. Yih-Chuan Lin and Shen-Chuan Tai "Fast Full-Search Block-Matching Algorithm for Motion-Compensated Video Compression" IEEE Transactions On Communications, Vol. 45, No. 5, May 1997 527.

12. J. S. Kim and R. H. Park, "A fast feature-based block matching algorithm using integral projections," IEEE J. Select. Areas Commun., vol. 10, pp. 968–971, June 1992.

13. Y. H. Fok and O. C. Au, "An improved fast feature-based block motion estimation," in Proc. IEEE 1994 Int. Conf. Image Processing, 1994, pp. 741–745.

14. H. B. Park and C. Wang, "Image compression by vector quantization of projection data," IEICE Trans. Inform. Syst., vol. E75-D, pp. 148–155, Jan. 1992.

15. K. H. Jung and C. Wang, "Projective image representation and its application to image compression," IEICE Trans. Inform. Syst., vol. E79-D, pp. 136–142, Feb. 1996.

16. MVC Software Reference Manual JMVC 8.2,ISO/IECMPEG,ITU-T VCEG, May 2010.

17. N. Purnachand, L .N. Alves, and A. Navzrro, "Fastmotion estimation algorithmforHEVC," in Proc .IEEE Int. Conf .Consum .Electron.,Sep. 2012, pp. 34–37.

18. C.-M. Kuo, Y.-H. Kuan, C.-H. Hsieh, and Y.-H. Lee, "A novel prediction-based directional asymmetric search algorithm for fast block matching motion estimation," IEEE Trans. Circuits Syst. Video Technol., vol. 19, no. 6, pp. 893–898, Jun. 2009.
19. W. Zhu, W. Ding, J. Xu, Y. Shi, and B. Yin, "2-D dictionary based video coding for screen contents," in Proc. Data Compression Conf., 2014, pp. 43–52.
20. A. D. Houghton, "Cyclic redundancy checking," in The Engineer's Error Coding Handbook. New York, NY, USA: Springer, 1996, pp. 12–24 IEEE Transactions On Multimedia, Vol.17,No.7,July2015.