DATA REUSE IN TWO-LEVEL HIERARCHICAL MOTION ESTIMATION FOR HIGH RESOLUTION VIDEO CODING

Mariusz Jakubowski and Grzegorz Pastuszak
Institute of Radioelectronics, Warsaw University of Technology, ul. Nowowiejska 15/19, Warsaw, Poland

Keywords: Data Reuse, Memory Bandwidth, Motion Estimation, Video Coding.

Abstract: In hardware implementation of a video coding system memory access becomes a critical issue and the motion estimation (ME) module is the one which consumes most of the data access. To meet the requirements of HD specification, conventional data reuse schemes are not sufficient. In this paper, the hierarchical approach to ME is combined with the Level C and D data reuse schemes. Proposed two-level hierarchical ME algorithm reduces the external memory bandwidth by 77% and the on-chip memory size by 93% with reference to the Level C scheme, and computational complexity by over 99% with reference to the one-level full search (OLFS), achieving the results close to OLFS.

1 INTRODUCTION

Motion estimation (ME) is a key element of standard video coders such as H.26x, MPEG-1, MPEG-2, and MPEG-4, which reduces temporal redundancies existing in video sequences. The most popular method, block-matching ME, relies on searching the best matching position of a 16×16 pixels or smaller block from a current frame within a predetermined or adaptive search range (SR) in a reference frame. In hardware implementations, the most popular ME algorithm is full search (FS) which checks all the positions within the search window. FS is regular and easy to control but very expensive in terms of computational load and memory transfer. Due to the rapid advances in VLSI technology, computational complexity requirements became less critical than available memory bandwidth. To reduce the data transfer from the external memory, four levels of search area data reuse from A to D can be distinguished (Tuan *et al.*, 2001). Each of the levels exploits overlapping of adjacent candidate blocks, candidate block strips, search areas, and search area strips, respectively. The Level C scheme is often selected as offering substantial reduction of external memory bandwidth (EMB) with a reasonable size of the on-chip memory. Even with the Level C scheme applied to HD720p video (1280×720, 30 fps) with SR [-128, 128) and one reference frame, the required memory bandwidth is reduced from 1.7 TB/s to 536 MB/s, which is still a large number.

To get further reduction of EMB and speed up the ME process at the same time, two-level hierarchical search (TLHS) algorithm is proposed. Different types of the hierarchical search are quite popular in hardware implementations of the ME process (Mizosoe *et al*., 2007), (Chang *et al.*, 2009). Hierarchical search is based on the idea of searching the best match on the coarse level of subsampled search area (SA) and use it as the starting point for the refinement on the finer level. Since the number of points to check and the size of SA on each level are highly reduced, both the computational complexity and EMB are substantially lower than in case of the one-level search. In the proposed solution, it has been decided to use two-level search scheme as the most beneficial from the data reuse point of view. Due to large SRs used for ME in HD sequences, the first-level search is performed on the subsampled image with 4:1 ratio in both directions and the second-level search at the full resolution within a much smaller SR. The results of experiments performed on several HD sequences show that performance of the proposed solution is close to achieved by one-level FS (OLFS) at the full-resolution image with 77% of EMB reduction with reference to Level C and over 99% reduction of computational complexity with reference to OLFS.

The rest of the paper is organized as follows. In Section II, possibilities of the memory bandwidth reduction in FS ME algorithm are presented, and the proposed TLHS scheme is introduced together with the estimation of the EMB and computational complexity reduction. In Section III, the results of the experiments on a few HD720p sequences are

Jakubowski M. and Pastuszak G. (2010).
DATA REUSE IN TWO-LEVEL HIERARCHICAL MOTION ESTIMATION FOR HIGH RESOLUTION VIDEO CODING.
In *Proceedings of the International Conference on Signal Processing and Multimedia Applications*, pages 159-162
DOI: 10.5220/0002991801590162

presented. Section IV gives a conclusion.

2 DATA REUSE IN TWO-LEVEL HIERARCHICAL MOTION ESTIMATION

2.1 Data Reuse Schemes for Full Search Motion Estimation

FS ME algorithm checks each candidate inside SA. If the horizontal SR is [$-p_H$, p_H), the vertical SR [$-p_V$, p_V), and the size of the block $N \times N$, the number of positions to check is $4p_H \times p_V$ and the size of SA is $(2p_H + N - 1) \times (2p_V + N - 1)$. Since the adjacent candidate blocks inside SA and SAs of adjacent current blocks are highly overlapped, it creates the opportunity for effective data reuse and EMB reduction at the expense of the on-chip memory size increase.

In (Tuan *et al.*, 2002), four levels of the data reuse have been distinguished from Level A to D. The higher the level, the larger EMB reduction, however, the larger the size of the on-chip memory at the same time. Level A reuses pixels of two horizontally adjacent candidate blocks; Level B, pixels of two vertically overlapped candidate block strips; Level C, pixels of two horizontally overlapped SAs of two adjacent current blocks; and Level D, pixels of two vertically overlapped SA strips. The most popular scheme of Level C is presented in Fig. 1. With HD1080p video, 30 fps, N = 16, and [-192, 192)× [-128, 128) SA, requirements for the on-chip memory size and EMB are 101 kB and 1.17 GB/s at Level C, and 574 kB and 60 MB/s at Level D, assuming eight-bit-pixel precision. It is clear that the implementation any of these data reuse schemes might be too costly either for the sake of EMB or the on-chip memory size, and further reduction of these parameters is necessary.

2.2 Two-level Hierarchical Search ME Algorithm

Hierarchical search is quite popular approach used in many VLSI architectures to reduce the computational complexity of ME. Usually, two or three levels of hierarchy are used, and MV found on the higher (coarse) level becomes the search center for the next (finer) level. The size of a current block is maintained fixed or increased on each level. When size of a current block is kept constant on each level, an initial MV is obtained from a relatively large area which makes it less noise sensitive but often also less accurate as a larger block on the coarse level covers a few blocks on the finer level. Thus, in the proposed solution, it has been decided to scale the current block size according to subsampling of the reference frame.

Regarding the number of search levels, two levels of hierarchy are most beneficial from the data reuse point of view, since only on the first level SAs of adjacent blocks are overlapped. In general, SAs on the fine level are disjointed and have to be fetched from memory separately for each current block. Thus, any additional level of hierarchy increases EMB.

To create the coarse-level image, two approaches have been considered: subsampling with 4:1 factor in each direction, and low-pass filtering by simple averaging. Subsampling is accomplished by the selection of one pixel from each 16×16 block of a reference frame and does not required any extra computation or memory, however, the presence of noise might deteriorate an initial MV estimation. On the other hand, averaging of a 16×16 block requires 15 additions and one division by 16 (which can be easily accomplished by shifting) and the averaged image must be prepared in advance and stored in an external memory. With the noise reduction, a more accurate estimation of an initial MV might be expected.

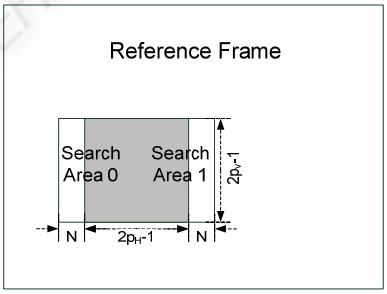

Figure 1: Level C data reuse scheme for FS algorithm. Overlapped and reused area is grey coloured.

On the first search level of hierarchy, FS ME is performed on a subsampled or averaged SA. On the next level, the refinement of an initial MV found on the previous stage is performed on the full-resolution SA but much smaller than an initial one. The initial experiments with five CIF sequences (Container, Football, Foreman, Mobile, News), 150 frames each and H.264/AVC JM12.0 reference software with SR

+/-32, quantization parameter QP = 25, one reference frame, variable block size, and quarter-pixel ME, allowed to determine that the refinement range +/-8 is sufficient to achieve the performance close to OLFS both for low- and high-motion activity sequences. The averaged SA gives better

Table 1: EMB and on-chip memory (OCM) size with Level C and D data reuse scheme for OLFS and TLHS.

Format	Level C - OLFS		Level D - OLFS		Level C - TLHS		Level D - TLHS	
	EMB [MB/s]	OCM [kB]	EMB [MB/s]	OCM [kB]	EMB [MB/s]	OCM [kB]	EMB [MB/s]	OCM [kB]
W = 1280, H = 720 FPS = 30, N = 16 p_H = 128, p_V = 128 p_{HR} = 8, p_{VR} = 8	536	67	26	382	122	5	101	25
W = 1920, H = 1080 FPS = 30, N = 16 p_H = 192, p_V = 128 p_{HR} = 8, p_{VR} = 8	1205	101	59	574	297	8	226	36

results especially for complex-texture sequences such as Mobile, where the noise influence is particularly harmful for ME efficiency.

2.3 EMB and Computation Complexity Reduction

Since block size, SR and frame size are reduced by the factor of 4 in each direction, the proposed solution offers 16 times reduction of the on-chip memory and from 16 up to 64 times reduction of the memory bandwidth with reference to OLFS. However, some additional on-chip memory and transfer are necessary for the second-level search. In particular, $(2p_{HR} + N - 1)\times (2p_{VR} + N - 1)$ pixels must be transferred and stored for each current block, where p_{HR} and p_{VR} is the refinement range in horizontal and vertical direction, respectively. In Table 1, the values of EMB and on-chip memory (OCM) size for Level C and D with OLFS and TLHS have been compared for HD720p and HD1080p sequences, assuming eight-bit-pixel precision and one reference frame. For the Level C scheme with TLHS, EMB is reduced over 4 times and OCM size about 13 times with reference to OLFS. In case of the Level D scheme with TLHS, EMB has been increased almost 4 times but OCM size reduced over 15 times with reference to OLFS, which makes it reasonable to exploit this level in an actual VLSI implementation. If necessary, further EMB reduction can be obtained with smaller refinement SR at the expense of coding efficiency.

Regarding the computational complexity reduction, on the coarse level, 16 times less positions have to be checked in comparison with OLFS. Additionally, since size of a block is reduced 16 times, instead of 767 operations for sum of absolute differences (SAD) calculation, only 47 operations per single search point are necessary. If averaged SA is used, 15 additions and one division per pixel of the reference image are necessary but amount of these operations per reference frame is negligible in comparison with the rest of computation.

On the fine level, if FS is used as the search method, $4p_{HR}\times p_{VR}$ positions have to be checked. To get further reduction of the computational complexity on the fine level, instead FS, three-step search (TSS) algorithm can be considered as combining regularity with a reasonable performance. Number of search points required for TSS equals to $1 + 8log_2p_{HR}$. When variable block size and sub-pel ME are used, additional operations for composing SADs for larger blocks and generating fractional-pel positions are necessary but they are not taken into account in this analysis.

Table 2: OLFS and TLHS computational load comparison.

Computational load [GOPS] Format	OLFS	TLHS	
		FS on the fine level	TSS on the fine level
W = 1280, H = 720 FPS = 30, N = 16 p_H = 128, p_V = 128 p_{HR} = 8, p_{VR} = 8	5056	42	23
W = 1920, H = 1080 FPS = 30, N = 16 p_H = 192, p_V = 128 p_{HR} = 8, p_{VR} = 8	17064	118	75

In Table 2, the computational load required for OLFS and THLS with FS and TSS algorithm on the fine level is presented. Obtained reduction of the computational load is about 99.2% and 99.55% when FS and TSS are used on the fine level, respectively. In the next section results of experiments with HD sequences show the impact of the proposed solution on the coding efficiency.

3 EXPERIMENTAL RESULTS

In the experiments, OLFS is compared with TLHS

Table 3: OLFS and TLHS performance comparison.

Algorithm	OLFS		TLHS							
			Subsampled search area				Averaged search area			
			FS on the fine level		TSS on the fine level		FS on the fine level		TSS on the fine level	
Sequence	BR [kb/s]	PSNR [dB]	BR [kb/s]	PSNR [dB]	BR [kb/s]	PSNR [dB]	BR [kb/s]	PSNR [dB]	BR [kb/s]	PSNR [dB]
MobCal	26390.27	37.647	26434.04	37.531	26714.09	37.460	26297.71	37.648	26378.39	37.635
ParkRun	65270.77	37.574	66190.96	37.579	66435.46	37.581	65229.15	37.575	65230.43	37.574
Shields	21041.15	37.781	21156.13	36.794	25173.03	36.612	20924.13	37.071	23692.43	36.843
Stockholm	27522.66	37.628	27531.78	37.617	27708.55	37.614	27541.68	37.630	27655.61	37.633

using both subampled and averaged SA on the coarse level and FS and TSS ME algorithm on the fine level. Four HD720p sequences were used, 150 frames each: MobCal, ParkRun, Shields, and Stockholm. SR was set at 128 both in horizontal and vertical direction, QP was set at 25, the GOP structure was IPPP and one reference frame was used with the variable block size and sub-pel ME turned on. On the fine level, SR value was set at 8. Obtained bitrate and PSNR values for each algorithm are presented in Table 3.

In most cases, TLHS performance is close to OLFS. In particular, when averaged SA and FS on the fine level are used, the bitrate is even lower than obtained with OLFS. It may be explained by the fact that low-pass filtered image is more suitable for true motion detection and MV found during the integer-pel ME gives a better starting point for the sub-pel ME. In general, averaged SA gives better results both with FS and TSS on the fine level. The only sequence for which the TLHS performs visibly worse than OLFS is Shields. With the aveaged SA and FS used on the fine level the difference in PSNR equals to 0.71 dB with a slightly lower bitrate, but for the rest of settings, the difference in PSNR exceeds 1 dB and the bitrate might be even 20% higher than in case of OLFS. In this sequence, camera pans slowly across a wall of colourful shields. Due to plenty of fine details in this sequence, when over 90% of pixels are removed from an image by subsampling, an initial MV estimation might be inaccurate. The increase of the fine-level SA seems not to be the solution, since the refinement range equals to 40 is required to reduce the difference below 0.4 dB. Still, considering the huge EMB and computational load reduction, this deterioration of quality can be accepted. For the rest of the sequences, the difference in PSNR is 0.2 dB at most and less than 2% in bitrate.

4 CONCLUSIONS

Presented two-level hierarchical search ME algorithm combined with conventional data reuse schemes reduces EMB up to 77% and the computational load over 99% with reference to OLFS. The first-level search is performed on the subsampled image while search on the fine level - on the full-resolution image within a much smaller SR. Results of the experiments performed on a few HD sequences show that TLHS performance, in most cases, is almost the same as OLFS.

ACKNOWLEDGEMENTS

The work presented was developed within the Polish government's Innovative Economy Operational Programme 2007-2013.

REFERENCES

Tuan, J. C., Chang, T. S., Jen, C. W., 2002. On the data reuse and memory bandwidth analysis for full-search block-matching VLSI architecture. *IEEE Trans. Circuits Syst. Video Technol.*, vol.12, no.1, pp.61-72.

Mizosoe, H., Yoshida, D., Nakamura, T., 2007. A Single Chip H.264/AVC HDTV Encoder/Decoder/Transco der System LSI. " *IEEE Trans. Consumer Electronics,* vol.53, no.2, pp.630-635.

Chang, H. C. , Chen, J. W. , Wu, B. T., Su, C. L., Wang, J. S., Guo, J. I., 2009. A Dynamic Quality-Adjustable H.264 Video Encoder for Power-Aware Video Applications. *IEEE Trans. Circuits Syst. Video Technol.*, vol.19, no.12, pp.1739-1754.

Koga, T., Iinuma, K., Hirano, A., Iijima, Y., and Ishiguro, T., 1981. Motion Compensated Interframe Coding for Video Conferencing. In *Proc. Nat. Telecom. Conf.*, pp. C9.6.1–C9.6.5.