Signal Processing: *Image Communication* 31 (2015) 134–150

Contents lists available at ScienceDirect

Signal Processing: *Image Communication*

journal homepage: www.elsevier.com/locate/image

Computation-constrained dynamic search range control for real-time video encoder

Xianghu Ji, Huizhu Jia*, Jie Liu, Xiaodong Xie, Wen Gao

National Engineering Laboratory for Video Technology, Peking University, Beijing 100871, China

ARTICLE INFO

Article history:
Received 26 March 2014
Received in revised form
12 November 2014
Accepted 5 December 2014
Available online 29 December 2014

Keywords:
Video coding
Motion estimation
Dynamic search range
VLSI architecture

ABSTRACT

Search range (SR) is a key parameter on the search quality control for motion estimation (ME) of a real-time video encoder. Dynamic search range (DSR) is a commonly employed algorithm to reduce the computational complexity of ME in a video encoder. In this paper, we model an effective predicted motion vector (PMV) deviation metric to predict the relationship between SR and motion vector difference (MVD), according to the prediction differences of both temporal and spatial motions of neighboring blocks. In addition, a computation-constrained DSR (CDSR) control algorithm is proposed to manage the computational complexity while maximizing video coding quality in a real-time computational constrained scenario. The SR is dynamically determined by three factors: motion complexity, user-defined probability and computation budget. Compared to the conventional DSR algorithms, the proposed CDSR is an effective and quantifiable algorithm to allocate more computation budget to the blocks with high PMV deviations (such as motion object boundary), and less computation budget to the well-matched motion predicted blocks, while maintaining a constrained computation requirement. Experimental results show that the proposed CDSR control algorithm is an effective method to manage the computation consumption of the DSR algorithm while keeping similar rate–distortion (RD) performance. It can achieve about 0.1–0.3 dB average PSNR improvement when the computation consumption is restricted to a specific level as compared with its equivalent Fixed SR algorithm and can achieve about 50–90% computation savings when compared to the benchmarks. For ME with high performance Processing Element (PE) engine, the quality degradation caused by the proposed CDSR algorithm can be ignored.

1. Introduction

In the hybrid block-based video coding standards such as ITU-T H.26x [1,2] and ISO/IEC MPEG-x series [3], motion estimation (ME) plays a vital role in achieving high compression efficiency by removing temporal redundancy between successive video frames. Motion estimation (ME) is defined as the process of searching for an optimal motion vector (MV) that represents displacement of coordinates of the best matched block in a reference frame (past/future frame) for the block in the current frame. Motion vector prediction that predicts motion vectors by exploiting the correlation of MVs between spatial or temporal neighboring blocks [4,5] is a popular technique used in various video coding standards, and the predicted motion vector (PMV) is often adopted as the start point for many ME algorithms. For the window based ME algorithms, an area in the reference frame within the predefined search range around the start point (the

* Corresponding author at: Peking University, No. 5 Yiheyuan Road Haidian District, Beijing 100871, China. Tel.: +86 18910958581.
E-mail addresses: xhji@jdl.ac.cn (X. Ji), hzjia@pku.edu.cn (H. Jia), jliu@jdl.ac.cn (J. Liu), xdxie@pku.edu.cn (X. Xie), wgao@pku.edu.cn (W. Gao).

http://dx.doi.org/10.1016/j.image.2014.12.002
0923-5965/

collocated block or the PMV) is defined as the search window (SW). Then searches are conducted on the candidates within a SW of a reference frame. The ME calculates the matching costs of the candidates in the SW, and the candidate with the smallest matching cost is the best match. The most common criterion of the matching cost is the sum of absolute differences (SAD) between pixels of the current block and pixels of the reference candidate.

With the applications of more efficient coding techniques and the increasing requirement of higher video resolutions, the computation complexity of ME has been dramatically increased. Studies have been done and shown that more than 70% of the total encoding time and 90% of the total memory access are dedicated to the ME process [6]. Therefore, many fast algorithms have been proposed to reduce the computational complexity of ME. Among all the fast algorithms, search point reduction is a very straightforward and effective method [7] to accelerate the ME process. For one category of those search point reducing algorithms, such as the three-step search [8], the four-step search [9], the diamond search [10] and the hexagon-based search [11], search points reduction is achieved by applying inherent search patterns. Although those fast algorithms alleviate the problem of high computational complexity, they are usually not hardware friendly and suffer from the problems of performance degradation, unpredictable memory access or irregular data flow, etc. For the ME algorithms (full search, hierarchical search, etc.) in which searching points are mainly determined by the search window, SR reduction algorithms are another way to reduce the computational complexity of ME. Dynamic search range (DSR) algorithms have been proposed by researchers [12–21] recently. The proposed DSR algorithm dynamically adjusts the SR for each Macroblock (MB) according to the information given by previously encoded syntax element. Hong [12] proposed a DSR algorithm using the motion vector (MV) information of the adjacent and previously coded blocks. The predicted SR of the current MB was determined by the magnitude of the MVs of neighboring blocks. Xu [13] improved the algorithm proposed in [12] to make it more suitable for H.264/AVC [2]. Yamada [14] presented a two-stage SR modification algorithm that limits the SR by the MVs of neighboring blocks as well as the prediction error of the corresponding block in the reference frame. Zhang [15] determined the SR based on the frame complexity measured by the degree of motion activity at frame level. Shimizus [16] utilized the information of variable block size in H.264/AVC. He performed 16×16 mode or 8×8 mode at first and utilized the resulting motion vectors to reduce the SR for large block modes. Song [17] evaluated the relationship between SR and average MVs. The optimal SR for the current MB was determined based on the estimated average MVs and a pre-defined threshold. Ko [18] and Dai [19] used different zero mean 2-D distributions to model the probability density function (PDF) of MVD and the neighboring MVD information was adopted to estimate the distribution of the current MVD which determines the optimal SR. Lou [20] proposed a DSR algorithm by using both spatial and temporal MVPs as parameters to adaptively select the SR. Afterwards, the algorithm was further improved in [21] by using the variance of the MVP set which has been proved to be highly correlated with the SR value.

However, most of the previous works focus on the reduction of the SR while keeping similar rate–distortion (RD) performance, ignoring the constraints of implementation. In real-time video application, ME is mostly implemented as a dedicated hardware accelerator [22–24] in which processing is guaranteed to finish within strict time constraint to permit search for the best matching block. To achieve the highest degree of parallelism, pipelined architectures [25,26] are the most commonly used techniques for hardware accelerators. The pipeline is designed to maximize the system throughput while satisfying latency and resource constraints. In addition, in order to improve the pipeline performance, all the pipeline stages are expected to have a similar processing time. For a fixed time budget, searching through a larger SW usually requires a high level of parallelism and results in large silicon area consumption for the ME stage. Furthermore, the DSR algorithms usually give an unrestricted dynamic SR for each MB which leads to a large variation of the processing time for ME. The computing resources required to guarantee real-time processing must be significantly increased with the search range and frame size to an unacceptable degree. Meanwhile, it is difficult to estimate the scale of computing resources for the DSR ME accelerator in order to guarantee the real-time processing of the system. For this reason, the dynamic SR that leads to an unpredictable computing requirement for ME in conventional DSR algorithms is not desirable for the hardware implementation. Hence, practical hardware-friendly DSR algorithm should not only consider the trade-off between the SR reduction and RD performance, but also its implications on hardware implementation.

In this paper, in order to overcome the problems of the conventional DSR algorithms, we propose an effective PMV deviation metric to model the distribution of the relationship between SR and MVD according to the prediction differences of both temporal and spatial neighboring motions. Furthermore, a computation-constrained DSR control algorithm is proposed to manage the computational complexity while maximizing video coding quality in a real-time computational constrained scenario. The SR is determined by three factors: motion complexity, user-defined probability (the probability of the optimal MV to fall into the desired window) and computation budget. The computation-constrained DSR control algorithm can be easily adapted to industrial products of real-time video encoders, such as live TV broadcasting, video telephone, particularly the VLSI based solution.

The remainder of this paper is organized as follows. The proposed Dynamic Search Range (DSR) prediction algorithm is presented in Section 2. In addition, a novel PMV deviation metric is introduced to model the distribution of the relationship between SR and MVD. In Section 3, the computation-constrained DSR control algorithm is presented. Experimental results of comparisons with the traditional non-constrained DSR algorithms are shown in Section 4. Finally, some concluding remarks are given in Section 5.