Computation-Aware Motion Estimation Algorithm Based On QoE Control

Zhou Bing, Zhou Yanjie, Xu Mingliang, Wu Fei
School of Information Engineering
Zhengzhou University
Zheng zhou,China
iebzhou@zzu.edu.cn
y.j.zhou.g@gmail.com
iexumingliang@zzu.edu.cn
wufei1423@163.com

***Abstract*—For the real-time video encoding on mobile devices, a QoE calculation model with two experience dimensions is designed in this paper, so users can select and adjust between video quality and duration. The QoE calculation model is used in the proposed novel one-pass computation-aware motion estimation algorithm in which blocks are divided into four categories. The computation which a frame can use is divided into two parts of the base layer and the gain layer, and then is allocated to four types of blocks respectively, meanwhile early termination detection and adaptive computation allocation strategy are taken in the algorithm. Experimental results show that our algorithm can more accurately to allocate computation and the QoE calculation model has practicality.**

Keywords—distortion gain; motion estimation; computation allocation; computation-aware; QoE;

I. Introduction

Motion estimation is the most time consuming function module in video encoding, and has been always studied as a key part[1]. When the power consumption and processing capability of mobile devices vary in a mobile application environment, the focus of the complexity control of the encoder is how to control the complexity of the motion estimation. With the launch of the new generation video coding standard of the HEVC[2], this problem becomes more prominent.

Traditional motion estimation algorithms, such as FSBMA, TSS[3], FSS[4], and DS[5], have one common characteristic: if there is no sufficient amount of computation, this will result in that the motion estimation process of a frame cannot be completed, and that not all the blocks in a frame can be processed. This feature causes that they are unable to adapt to computation-limited and computation-varied mobile application environments. Tai et al.[6] first proposed the concept of computation-aware estimation in which all the blocks in a frame are processed simultaneously and the block with the largest matching error is selected for motion estimation every time. But it is not always effective when computation is allocated to the block with the largest matching error. Yang et al.[7] established a priority for each search step of all the blocks and allocated computation according to the priority. Both methods process all the blocks in a multi-pass flow and cannot use the information from adjacent blocks. Huang et al.[8] proposed a one-pass program which can take advantage of the adjacent block information with a adaptive search strategy. Lin et al.[9] also used a one-pass program, which divides blocks into three categories combined with the motion vector information of a corresponding block in the previous frame and processes the current frame according to the classification information of the previous frame. The non-validity of the large-grained reference information can easily lead to a wide range of impact on coding quality.

Statistical results show that users' concern of mobile devices may include video quality and duration, but these algorithms just consider the video quality. In this paper, a QoE [10] calculation model is designed to guide the conduct of the motion estimation process in order to provide a better user experience that allows users to select and adjust between video quality and duration, and we propose a novel one-pass computation-aware motion estimation algorithm with the QoE model integrated.

II. Concept of Computation-Aware

The concept of computation-aware was first proposed by Tai et al.[6], and will be described with a four-block demonstration example. Figure 1(a) shows the search process of TTS, and the value of each node in the circle is the smallest matching error after each search step. For TSS, each block requires 25 search points, for a total of 100, and the total minimum compensated error is 170. If there is a sufficient amount of computation for at least 100 search points, TSS will perform well. If there is not a sufficient amount of computation, such as only 50 search points, the last two blocks cannot be processed for motion estimation, and the unprocessed block may lead to a very large coding distortion. In the computation-aware motion estimation, the computation that a frame can use is allocated to all the blocks by some kind of importance metric of blocks, so as to ensure that each block can be processed and the computation is used legitimately. Unfortunately, it is difficult to find an optimal solution to allocate the limited computation, because it is impossible to judge whether the matching error of a block is minimal, unless all subsequent search points are searched. Despite that it is hard to find a best solution, some heuristic methods still can be considered. The experimental results show that blocks with larger initial matching error are more likely to further reduce

978-1-4799-4811-6/14/$31.00

the matching error. That means the more amount of computation allocated to these blocks with a larger initial matching error, the closer can be made from the best solution. The strategy taken in Figure 1(b) is just like this, which requires a total of 36 search points and the total compensated error is 170.

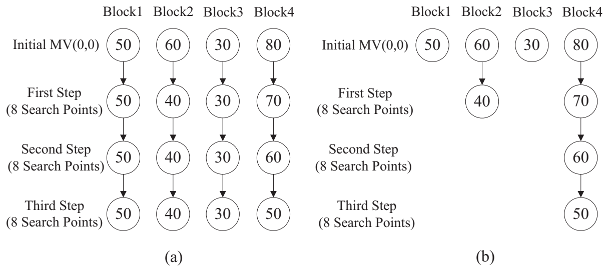

Fig. 1. Concept illustration: (a) traditional TSS (b) optimal TSS

III. DVFS Technique

DVFS is short for dynamic voltage and frequency scaling [11], which is used to adjust the clock frequency and supply voltage of the processor chip according to the different requirements and needs of applications running on it, so as to achieve the purpose of saving energy in the case of ensuring the required processing speed. Generally, the CPU power consumption P_{CPU} and other parameters have the following relationship:

$$P_{CPU} = \delta \cdot f_{clock} \cdot V^2 \quad (1)$$

where δ is a coefficient related to the CPU architecture, f_{clock} is the CPU clock frequency, V is the CPU supply voltage. Through the DVFS technique to adjust the CPU clock frequency, the energy saving can be obtained by the sacrifice of a part of the processing speed. The DVFS technique has a very wide range of applications, because it is able to save energy consumption, which is the first request for most mobile devices.

IV. QoE Calculation Model

For video applications on mobile devices, users usually have two experience dimension pursuits, namely video quality and duration, and the two experience dimensions are mutually contradictory, that means they need a balance between each other. In order to provide a better user experience, which allows users to participate in the formulation of the equilibrium point, and therefore a need for a QoE calculation model to guide the conduct of the entire process is necessary. The general process of establishing a QoE calculation model is: first find the parameters that affect the user experience, then collect the user experience evaluation, and finally establish a mapping relationship between QoE and these parameters. Establishment of the mapping relationship may use stochastic neural network[12], exponential relationship[13], or a logarithmic relationship [14]. The maximum value of a QoE calculation model lies in its real-time application. So if we just want some simple experience report, it is more effective to do some questionnaires directly. The QoE calculation model designed in this paper is used in the proposed novel computation-aware motion estimation algorithm, so that users can select and adjust between video quality and duration. In order to develop a benchmark between video quality and duration, this paper has chosen two parameters of mobile devices, namely the utilization of computation resource and power status, and when their values are the same is as the benchmark. $D_{experience}$ represents the user experience dimension pursuit:

$$D_{experience} = U_{computation} - S_{charge} \quad (2)$$

where $U_{computation}$ represents the utilization of computation resource of mobile devices, and S_{charge} represents the power status of mobile devices. In specific applications, $U_{computation}$ is the ratio of the current CPU clock frequency and the maximum CPU clock frequency, which can be easily adjusted through the DVFS technique. When $D_{experience}$ is greater than 0 indicates that the user experience dimension pursuit is video quality; when $D_{experience}$ is less than 0 indicates that the user experience dimension pursuit is video duration. In order to quantitatively describe the two experience dimensions, the QoE calculation model designed in this paper provides 100 numeric values for each experience dimension for users to choose, which is represented by S_{QoE}.When the user experience dimension pursuit is video quality, $U_{computation}$ will be adjusted to the percentage between 1 and the selected S_{QoE} through the DVFS technique; When the user experience dimension pursuit is video duration, $U_{computation}$ will be adjusted to the percentage between 0 and the selected S_{QoE} through the DVFS technique. The specific calculation process is as follows:

$$U_{computation} = \begin{cases} U_{computation}\,(1-\dfrac{S_{QoE}}{100}) + \dfrac{S_{QoE}}{100} & \text{if } D_{experience} \ge 0 \\ U_{computation}\,(1-\dfrac{S_{QoE}}{100}) & \text{if } D_{experience} < 0 \end{cases} \quad (3)$$

V. Algorithm Description

After determining the amount of computation that a frame can use according to the QoE calculation model, the next step is to allocate the computation to all the blocks. The proposed algorithm consists of two processes: frame-level computation allocation and block-level computation allocation.

There are two constraints in the frame-level computation allocation: one is smoothness constraint, another is efficiency constraint. The smoothness constraint is used to keep changes in quality among different video frames as small as possible. The efficiency constraint is used to terminate unnecessary searching in the search for candidate matching block. Because the block-level computation allocation algorithm ensures that the actual computation consumption will not exceed the allocated computation, each frame is allocated a fixed amount of computation in the simulation experiment in order to facilitate comparison with other methods.

In the block-level computation allocation, first blocks are divided into four categories, then the computation a frame can use is divided into two parts of the base layer and the gain layer, last the computation in the base layer and the gain layer is allocated to all the blocks of a frame according to the type of each block.

Let's first describe how to divide blocks into different categories, where the distortion gain information of the previous frame is used. Each block in the current frame has a prediction motion vector pointing to a block in the previous frame, and the pointed block corresponds to a distortion gain which is the difference between its initial matching error and final matching error. If the pointed block covers a number of blocks, then the distortion gain of the current block is a weighted sum of the distortion gain of all covered blocks. Be_{block} represents the distortion gain of the current block:

$$Be_{block} = \sum_i \omega_i \cdot Be_{b\_pre\_i} \ (i>0) \tag{4}$$

Where $Be_{b\_pre\_i}$ represents the distortion gain of covered block i, and ω_i is the percentage of the area covered by the corresponding block i. $Dif_{b\_initial}$ represents the initial matching error of a block, then the four types of blocks are defined as follows:

$$Class_{block} = \begin{cases} 1 \ \text{ if } Dif_{b\_initial} \le dths \\ 2 \ \text{ if } dths < Dif_{b\_initial} \le dthl \\ 3 \ \text{ if } Dif_{b\_initial} > dthl \text{ and } Be_{block} \le bth \\ 4 \ \text{ if } Dif_{b\_initial} > dthl \text{ and } Be_{block} > bth \end{cases} \tag{5}$$

Where $Class_{block}$ represents the current block category, $dths$ is the small threshold of the initial matching error of a block, $dthl$ is the large threshold of the initial matching error of a block, and bth is a separation point meaning whether the distortion gain of a block is large or small.

Let's then describe the details of the block-level computation allocation algorithm. In the block-level computation allocation, the computation which a frame can use is divided into two parts of the base layer and the gain layer, where in the proportion of the computation allocated to the base layer is α(0<α<1). The computation in the base layer is allocated to all the four types of blocks according to the initial matching error of each block. C_{b_base} represents the computation allocated to a block in the base layer:

$$C_{b\_base} = \frac{C_{base\_available}}{Num_{b\_nopro}} \cdot \frac{Dif_{b\_initial}}{AvgDif} \tag{6}$$

Where $C_{base\_available}$ represents the computation a frame can use in the base layer, Num_{b_nopro} represents the number of unprocessed blocks in a frame, and $AvgDif$ represents the average initial matching error of all the processed blocks in a frame. The computation in the gain layer is also allocated to all the four types of blocks according to the distortion gain of each block. $C_{b\_benefit}$ represents the computation allocated to a block in the gain layer:

$$C_{b\_benefit} = C_{benefit\_total} \cdot \frac{Be_{block}}{Be_{total\_pre}} \tag{7}$$

Where $C_{benefit\_total}$ represents the total computation for a frame is allocated in the gain layer, and Be_{total_pre} represents the total distortion gain of a frame.

In order to reduce the impact of the non-validity of the distortion gain information, when the best motion vector gotten in a search step of type 1 block and type 2 block is the prediction motion vector, their searching will be stopped to reduce the unnecessary use of computation, because now the block is likely to have reached the global optimum. When the distortion gain ratio in the last search step of type 2 block and type 3 block is greater than β(0<β<1), the distortion gain ratio is the percentage of the reduced amount of distortion, then more computation will be allocated in order to carry out one more step search. When the distortion gain ratio in a continuous two-step search of type 4 block is less than γ(0<γ<1), the searching will be immediately stopped.

Figure 2 shows the process frame of the algorithm. In order to make the search strategy more general, DS is used as the motion estimation algorithm in the experiment in which the prediction motion vector is used as the initial search point. In order to speed up the search process, some other traditional acceleration strategies are taken, for example, when part of the block matching error is greater than the minimum matching error, the calculation of the block matching error will be stopped.

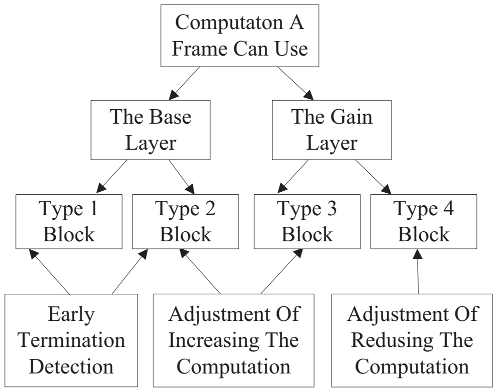

Fig. 2. The process frame of the algorithm

VI. SIMULATION RESULTS

Mobile, Bus and Highway of CIF format are selected as the test video sequences, since they contain complex irregular movement property, complex regular movement property and simple gentle movement property respectively. For all the sequences, the picture coding structure is IPPP..., the frame rate is 15Hz, the block size is 8*8, the quantization parameter is 28, the number of reference frames is 1, the search range is [-16,16], the control parameter α is set as 0.4, β and γ are both set as 0.1. The definition of the size of these two parameters *dths* and *dthl* in the experiment is: large if they are in the top 30% of the related information in the previous frame; small if they are in the bottom 30% of the related information in the previous frame. The value of *bth* is set to be the median of the related information in the previous frame. All computation allocation in the first P frame is just according to the initial matching error. In order to verify the efficiency of the proposed algorithm, the following comparison programs are designed: pro1 represents the proposed algorithm without QoE control, pro2 represents the proposed algorithm with QoE control, add1 represents the program that computation allocation is only according to the initial block matching error, add2 represents the program that computation allocation is according to the corresponding motion vector in the previous frame of the current block. In order to clearly present the experimental results, power decrement in the QoE control is in unit of frame, 1% of power per frame decrements, and the maximum amount of computation which a frame can use is 30000 search points. The QoE control program used in the experiment is: $D_{experience}$ is kept as 1, namely keep the utilization of computation resource of mobile devices equal to the power status of mobile devices. Each test video sequence was encoded 60 frames, and statistics were made from the second frame. Figure 3, 4 and 5 show the experimental results.

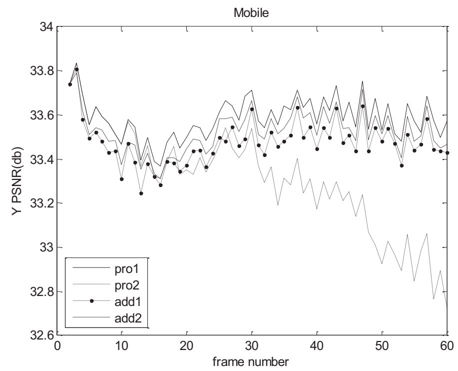

Fig. 3. Experimental results of Mobile sequence

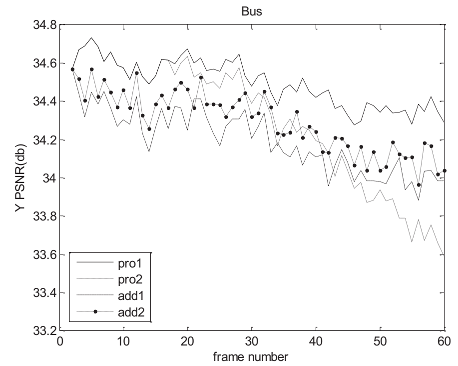

Fig. 4. Experimental results of Bus sequence

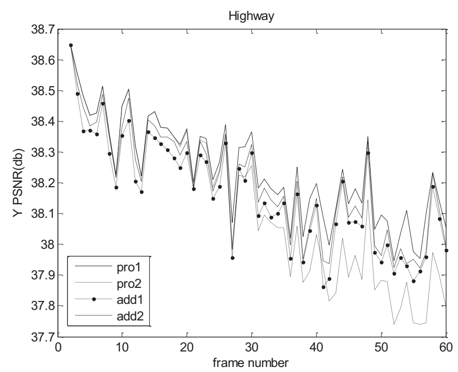

Fig. 5. Experimental results of Highway sequence

By comparing the experimental results of pro1, add1 and add2 on the three test video sequences. pro1 is more precise in computation allocation, thereby obtaining a higher video quality. By comparing the experimental results of pro1, add1 and add2 on the test video sequences of Mobile and Bus, the improvement of the former video quality by pro1 is not better than the latter's, but is better than those by add1 and by add2. This shows that, for complex irregular video sequences, the computation allocation using the distortion gain information is not so effective, but this can be compensated with an adaptive computation allocation strategy and early termination detection. For simple gentle video sequence, Highway, different algorithms have little difference. By comparing the experimental results of pro1 and pro2 on the three test video sequences, without reducing video quality, pro2 use less computation. This shows the efficiency of the proposed algorithm and the practicality of the QoE calculation model from another point of view.

VII. Conclusion

A QoE calculation model with two experience dimensions is designed in this paper, so that users can select and adjust between video quality and duration. The QoE calculation model is used in the proposed novel one-pass computation-aware motion estimation algorithm in which blocks are divided into four categories and the computation which a frame can use is divided into two parts of the base layer and the gain layer. The main concern of this paper is not only confined to maximize the utilization of the limited computation, but also take full advantage of the limited computation and provided a better user experience. The experimental results demonstrate the efficiency of the algorithm and the practicality of the QoE calculation model. The efficiency of this algorithm can be further improved if combined with other more effective search strategy or energy control program.

Acknowledgment

This work is supported by National Science Foundation of China(Grants 61379079 and 61472370 and 61202207). The authors would like to thank the anonymous reviewers for their valuable comments and suggestions.

References

[1] J. Lin, Y. Chen, Y. Tsai, Y. Huang, and S. Lei, "Motion vector coding techniques for HEVC," in Multimedia Signal Processing (MMSP), 2011 IEEE 13th International Workshop on, 2011, pp. 1-6.

[2] B. Li, G. J. Sullivan and J. Xu, "Compression performance of high efficiency video coding (HEVC) working draft 4," in Circuits and Systems (ISCAS), 2012 ,pp. 886-889.

[3] T. Koga, "Motion-compensated interframe coding for video conferencing," in Proc. NTC'81, 1981, pp. 5-3.

[4] L. Po and W. Ma, "A novel four-step search algorithm for fast block motion estimation," Circuits and Systems for Video Technology, IEEE Transactions on, vol. 6, pp. 313-317, 1996.

[5] S. Zhu and K. Ma, "A new diamond search algorithm for fast block-matching motion estimation," Image Processing, IEEE Transactions on, vol. 9, pp. 287-290, 2000.

[6] P. Tai, S. Huang, C. Liu, and J. Wang, "Computation-aware scheme for software-based block motion estimation," Circuits and Systems for Video Technology, IEEE Transactions on, vol. 13, pp. 901-913, 2003.

[7] Z. Yang, H. Cai and J. Li, "A framework for fine-granular computational-complexity scalable motion estimation [real-time video coding applications]," in Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on, 2005, pp. 5473-5476.

[8] C. Chen, Y. Huang, C. Lee, and L. Chen, "One-pass computation-aware motion estimation with adaptive search strategy," Multimedia, IEEE Transactions on, vol. 8, pp. 698-706, 2006.

[9] W. Lin, K. Panusopone, D. M. Baylon, and M. Sun, "A computation control motion estimation method for complexity-scalable video coding," Circuits and Systems for Video Technology, IEEE Transactions on, vol. 20, pp. 1533-1543, 2010.

[10] S. Bled, "Definition of Quality of Experience," ITU-T SG12, 2007, pp. 16-25.

[11] Z. He, Y. Liang, L. Chen, I. Ahmad, and D. Wu, "Power-rate-distortion analysis for wireless video communication under energy constraints," Circuits and Systems for Video Technology, IEEE Transactions on, vol. 15, pp. 645-658, 2005.

[12] K. D. Singh, A. Ksentini and B. Marienval, "Quality of experience measurement tool for SVC video coding," in Communications (ICC), 2011 IEEE International Conference on, 2011, pp. 1-5.

[13] M. Fiedler, T. Hossfeld and P. Tran-Gia, "A generic quantitative relationship between quality of experience and quality of service," Network, IEEE, vol. 24, pp. 36-41, 2010.

[14] A. M. Tourapis, O. C. Au and M. L. Liou, "Highly efficient predictive zonal algorithms for fast block-matching motion estimation," Circuits and Systems for Video Technology, IEEE Transactions on, vol.12, pp.934-947,2002.