

Spatial and Temporal Data Parallelization of Multi-view Video Encoding Algorithm

Yi Pang
Computer Science
Department
Tsinghua University
Beijing, China
pangy@mails.tsinghua.edu.cn

Lifeng Sun
Computer Science
Department
Tsinghua University
Beijing, China
sunlf@mail.tsinghua.edu.cn

Songliu Guo
Computer Science
Department
Tsinghua University
Beijing, China
guosongliu01@mails.tsinghua.edu.cn

Shiqiang Yang
Computer Science
Department
Tsinghua University
Beijing, China
yangshq@tsinghua.edu.cn

Abstract—*Multi-view Video coding technology is proposed to resolve the problem of huge data storage and transmission for free-view and 3D interactive video. How to support real time multi-view video encoding which has high computing complexity with sharply increased multi-view video data is essential. In this paper, we proposed a solution of Spatial and Temporal Data Parallelization for Multi-view Video encoding algorithm based on IBM Cell multiprocessor system using Selections of Optimal Theories & Methods. The performance of our tasks distributing scheme is eight times faster than the serial algorithm, speedup is notable.*

Keywords—*Multi-view Video Coding (MVC), multiprocessor, parallel, IBM Cell, Selections of Optimal Parallel Theories & Methods*

Topic area—*Multimedia: Method and System*

I. INTRODUCTION

With the development of internet and computing technology, the next generation of multimedia applications will be interactive and realistic. Multi-view video technology is shooting an object from different angles with multi-cameras, encoding the different angles, transmitting the data, decoding, and synthesizing a 3D result. And it resolves presentation, interaction, codec and transmission of 3D interactive video issues. Multi-view video has fueled the rapid expansion of applications into free viewpoint video [1], three dimensional displays [2], [3], and high performance imaging [4]. Figure 1 shows three main schemes for cameras configuration.

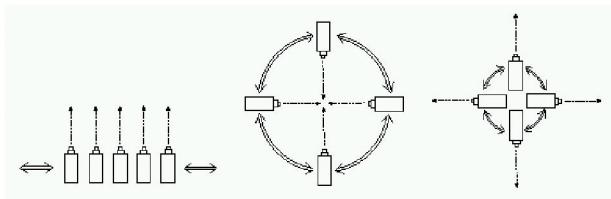


Figure 1 Configuration of Multi-view Video cameras

Multi-view Video Coding (MVC) standard is currently being established. The MPEG Committee has recently issued a Call for Proposals on Multi-view Video Coding [6] and has indicated that the final MVC standard will be an extension of H.264 and is scheduled to be released in 2008. More explanation of Multi-view Video Coding algorithm follows in Section II.

As the application coverage of these standards increases, it becomes apparent that real-time coding is more and more essential. Data and computing complexity of Multi-view Video Coding increase linearly, as the number of views increases. It is difficult for single core processor to support real time Multi-view Video Coding processing.

With the development of IC technology, the performance of transistors' speed increases and the physical dimensions of the circuits downsize. This larger processing capacity can be exploited to make multiprocessor on one chip. Chip multiprocessor (CMP) is a more power-efficient design due to the lower operation frequency and provides scalability over product ranges. CMP solution is imported by porting eight-view video coding algorithm onto IBM's Cell processor which is a nine-core processor. Our parallel MVC algorithm outperforms the serial algorithm by a factor of eight.

The rest of this paper is organized as follows, Section II introduces characteristics of MVC, Section III reviews multiprocessor architecture; Section IV covers decomposition and mapping method of MVC for multiprocessor architecture; and Section V is the conclusion.

II. MULTI-VIEW VIDEO CODING

High performance Multi-view Video Coding architecture is very important for 3D video. At the encoding end, it needs to effectively minimize the bit-rate for transmission and storage using both temporal and spatial redundancy between multi-cameras. At the decoding end, MVC shall support random access in the time, view dimension, and to a spatial area in a picture [7].

Traditional coding technologies, such as motion compensation and view disparity compensation, are not efficient for binocular stereo video coding and multi-view video (with large view disparity). In the MVC algorithm, to improve the motion compensation we utilize “global motion compensation”, which incorporates motion compensation and view disparity compensation. Motion compensation uses time redundancy and view disparity compensation uses space redundancy. MVC inherits the algorithm for one frame processing from H.264/AVC, and just redesigns reference relation among inter-reference frames. [8] - [10]

Considering different motion estimation methods, all the frames are categorized into three types: I frame, P frame and B frame [3]. In multi-view video encoding, I frame only needs its own content, P frame depends on its previous frame besides itself, B frame depends on its previous frame and its later frame, same as in H.264. Take 8-view video as an example, 8-view makes eight streams of video. The dependency graphs of 8-view video coding algorithm is described in Joint Draft 2.0 on Multi-view Video Coding [15] and Joint Multi-view Video Model (JMVM) 3.0[14] as Figure 2

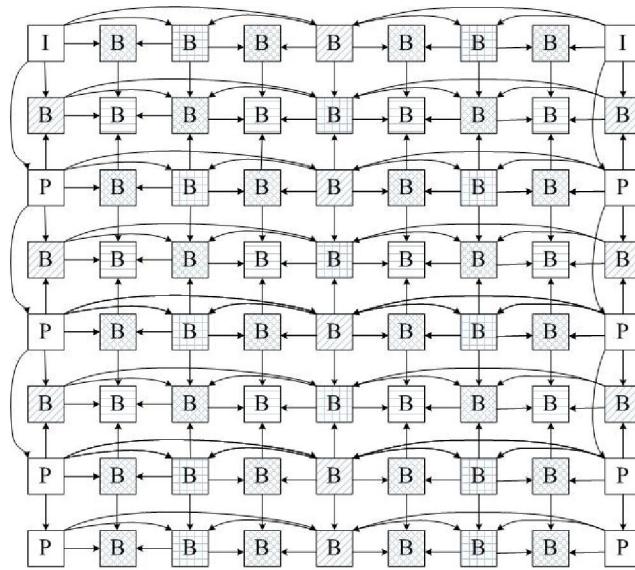


Figure 2 Dependency graphs of eight-view video coding for one GOP

In Figure 2 each row (S_0, S_1, \dots, S_7) represents a stream of video from a view, each column represents a time point, and each square is a process of encoding a frame. The letter on the square means that the square is either an I frame, P frame, or B frame process. B frames are sorted into three types: B2 refers to two frames, B3 refers to three frames, and B4 refers to four frames, which are determined by arrow direction. All the frame processes are distributed into six different priorities. The arrow's direction between different squares reveals the data dependency relationship. The direction of the arrow indicates the dependency. The square with the incoming arrow has dependency from the square with the outgoing arrow. In a Groups of Pictures

(GOP), the two I frames have the highest priority, they need to be encoded first; eight P frames have the second highest priority, then B frames have the next four priorities in the

following sequence B_{B2} s, the B_{B3} s, the B_{B4} s, and the B_{P} s. The lower priority frames can't be processed until higher priority frames have been done. The frames that have the same priority can be done in parallel. For each frame, MVC inherits H.264. Such 72 frames GOP have six different priority (P_1, P_2, \dots, P_6). All the frames are distributed in the following table:

TABLE 1 TABLE OF PARAMETERS

Frame type	The number of frames for each priority(from left to right, from high to low)						Quantity	Time for processing one frame
	P1	P2	P3	P4	P5	P6		
I	2	0	0	0	0	0	$N_I = 2$	T_I
P	0	8	0	0	0	0	$N_p = 8$	T_p
B2	0	0	10	8	16	0	$N_{B2} = 34$	T_{B2}
B3	0	0	1	2	4	0	$N_{B3} = 7$	T_{B3}
B4	0	0	0	3	6	12	$N_{B4} = 21$	T_{B4}
Total ($T_i, i=1, 2, \dots, 6$)	2	8	11	13	26	12	72	

Data of 8-view video is eight times larger than single-view videos, and the number of frames per second to be processed for real time increases from 30 to 240 [13].

The most important issue for MVC to resolve is single-core processors unable to complete high computing complexity for real-time processing, caused by data increasing sharply and more computing required, Porting MVC applications onto chip multiprocessors in parallel can solve this problem.

III. MULTIPROCESSOR ARCHITECTURES

Our solution of parallel multi-view video encoding algorithm explores the feasibility of mapping MVC algorithm onto a multiprocessor based on the architecture template described by Figure 3. This architecture consists of a Center Processor Element (CPE), multi Synergistic Processor Element (SPE), and communication infrastructures. All the SPUs are the same, and in charge of basic tasks. Each of them has a Local Storage (LS), all the code and data running on SPU are in its LS. CPE is the control center, controls data flow and schedules task order. Independently exchanging data or information between two SPUs is not allowed, CPE has to control it. All the data and code are transmitted by the BUS which has a wide bandwidth to transfer data quickly.

For the implementation of the parallel MVC algorithm, IBM Cell processor is assumed as the instantiation of the architecture template. It contains eight SPEs running at 4GHz with 32KB Instruction L1 Cache ,32KB Data L1 Cache and 512KB L2 Cache for PPE (Power Processor Element), 256KB Local Store for SPE, four 16 byte data rings supporting multiple simultaneous transfers per ring, 96Bytes/cycle peak bandwidth, over 100 outstanding requests for BUS. Besides these, there exist interfaces to off-chip (e.g. MIC, BIC, etc), as Figure 4. [12]

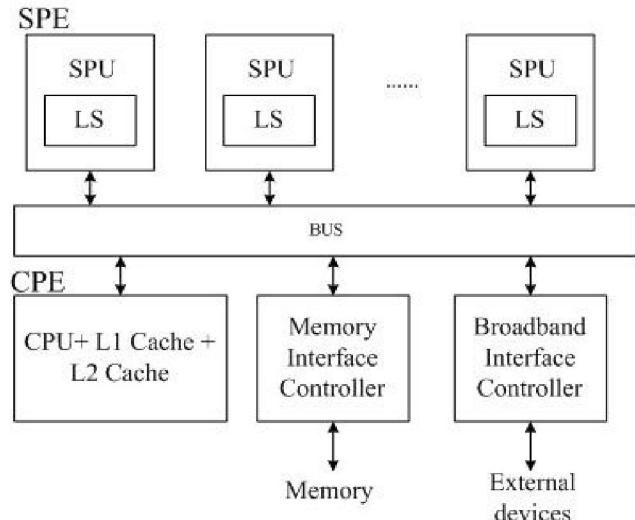


Figure 3 Target Multi-Processor Architecture

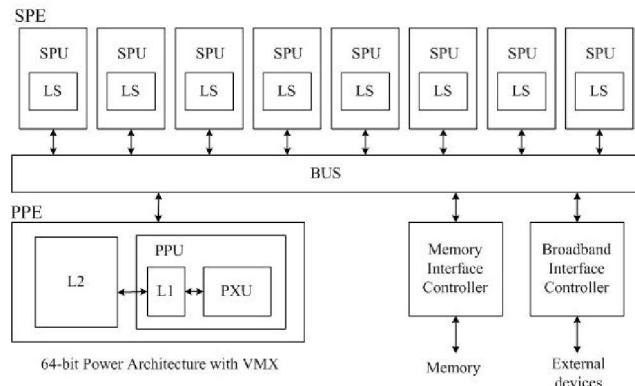


Figure 4 Cell Processor Architecture

IV. MULTI-VIEW VIDEO CODING PARTITIONING FOR MULTIPROCESSOR ARCHITECTURES

For parallel multi-view video processing, CMP could solve the large computing problem. Because of design complexity and power consuming, development of single core processor slows down. Next generation architecture is chip multiprocessor. The ease of programming of systems with a large amount of parallelism, the mapping of application software onto the system, is an important factor for business success, since the advantages of multiprocessor

architectures only hold when the parallelism can be exploited (Amdahl's law). Through analyzing Figure 2, parallelism of MVC is fit for multi-core processor. How to exploit parallelism for multi-core processor is what we focus on.

In video coding, frame is a kind of basic processing granularities, each frame process is distributed to a SPU. The processing order set by considering priority is controlled by PPE. Then nine processing units run in parallel. We get an arrangement referring to Selections of Optimal Theories and Methods [13]. Selections of Optimal Theories and Methods give us a way to find out a best solution of a problem, following some conditions conclude a maximum (or a minimum) of a parameter.

If the operation of each reference frame for motion estimation is the same, the computing of B frame referring to two frames is two more times than P frame which refers to one frame, so is the computing of B frame referring to four or three frames, and $T_1 < T_P < T_{B2} < T_{B3} < T_{B4}$.

T is the time for finishing the whole GOP encoding process, our goal is to make T as minimal as possible. Eight core is numbered 1st, 2nd, ..., 8th. Five kinds of frames are I frame, P frame, B2 frame, B3 frame and B4 frame. All the frames belong to six different priorities. N_{iCP_k} ($i=1,2,\dots,8$; $C=I,P,B2,B3,B4$; $k=1,2,\dots,6$) means number of frames which is C kind in k th priority running on the i th SPU. For example, the number of B2 frames whose priority is P3 processed on 3rd core is N_{B2P3} . Referring to Selections of Optimal Theories & Methods formulas follow:

Let T minimizes, and satisfies the following requirements

$$T = \sum_{k=1}^6 \max \left\{ \sum_{C=I,P,B2,B3,B4} N_{iCP_k} \times T_C, i=1,2,\dots,8 \right\} \quad (1)$$

$$T \geq \sum_{k=1}^6 \sum_{C=I,P,B2,B3,B4} N_{iCP_k} \times T_C, i=1,2,\dots,8 \quad (2)$$

$$\sum_{i=1}^8 \sum_{k=1}^6 N_{iCP_k} = N_C, C = I, P, B2, B3, B4 \quad (3)$$

$$\sum_{i=1}^8 \sum_{C=I,P,B2,B3,B4} N_{iCP_k} = T_k, k = 1, 2, \dots, 6 \quad (4)$$

$$0 < T_I < T_P < T_{B2} < T_{B3} < T_{B4}, \text{ all the parameters } \geq 0 \quad (5)$$

After combination and consideration of all formulas we get $T = T_I + T_P + 4T_{B2} + 2T_{B3} + 4T_{B4}$. In fact the last column is reused by the previous GOP and the latter GOP, so on average there are 64 frames in a GOP, 4 P frames in P2, 7

B2 frames and a B3 frame in P3. Therefore T_{Parallel} should be $T_1 + T_p + 3T_{B2} + 2T_{B3} + 4T_{B4}$. The task arrangement for encoding 64 frames matrix on eight-core processor shows as Figure 5

As Figure 5, y-axis is 1st-8th core, x-axis is time. Each line represents tasks arranged for each core, the whole chart means GOP of $8 \times 8 = 64$ frames distributed onto eight SPUs, and all the tasks are finished in order (from Priority 1 to Priority 6). Through Figure 5 all the N_{iCP_k} ($i=1,2,\dots,8$; $C=I,P,B2,B3,B4$; $k=1,2,\dots,6$) number can be looked up. For example, $N_{B2P5} = 2$, $N_{B3P5} = 0$. Each shadow bar is time to processing one frame.

For traditional serial algorithm, T_{serial} is $T_1 + 4T_p + 31T_{B2} + 7T_{B3} + 21T_{B4}$ which is 8 more times than T_{Parallel} .

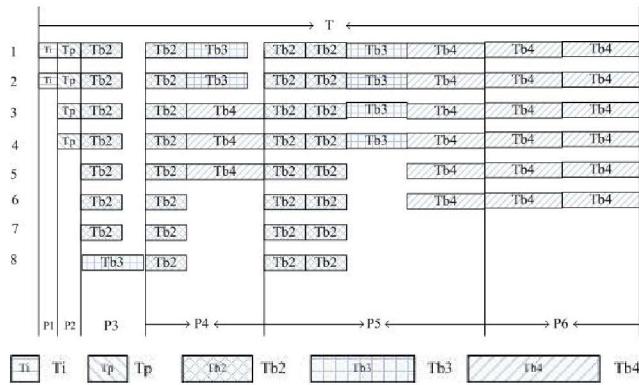


Figure 5 The optimal task schedule chart

No.60503063, 863 program under Grant No.2006AA01Z321 and Seed Fund of Tsinghua University.

REFERENCES

- [1] A. Smolic, P. Kauff, "Interactive 3-D video representation and coding technologies", *Proceedings of the IEEE*, vol. 93, no. 1, pp 98-110, Jan. 2005.
- [2] N. A. Dodgson, "Autostereoscopic 3D Displays", *IEEE Computer*, vol. 38, no. 8, pp. 31-36, Aug. 2005.
- [3] M. Tanimoto, "FTV (Free Viewpoint Television) Creating Ray-Based Image Engineering", *International Conference on Image Processing*, vol. 2, pp. 25-28, Sept. 2005.
- [4] B. Wilburn, et al., "High Performance Imaging Using Large Camera Arrays," *ACM Transactions on Graphics*, vol. 24, no. 3, pp. 765-776, July 2005.
- [5] Nelson H.C.Yung,Senior Member,IEEE, and Kwong-Keung Leung, Student Member IEEE Spatial and Temporal Data Parallelization of the H.261 Video Coding Algorithm *IEEE Transactions on Circuits and Systems for video Technology*, Vol.11,No.1, January 2001
- [6] ISO/IEC JTC1/SC29/WG11, "Updated Call for Proposals on Multi-view Video Coding", MPEG Doc. N7567, Nice, France, Oct. 2005.
- [7] ISO/IEC JTC1/SC29/WG11 N8218 "Requirements on Multi-view Video Coding v.7",July 2006
- [8] ISO/IEC JTC 1/SC 29/WG 11 N8459 "Joint Multi-view Video Model (JMVM) 2.0 ", Oct. 2006
- [9] ISO/IEC JTC 1/SC 29/WG 11 N8244 "Joint Multi-view Video Model (JMVM) 1.0", July 2006
- [10] Emin Martinian, Alexander Behrens, Jun Xin, Anthony Vetro, Huifang Sun "Extensions of H.264/AVC For Multi-view Video Compression" MITSUBISHI ELECTRIC RESEARCH LABORATORIES, May 2006
- [11] Erik B. van der Tol, Egbert G.T.Jaspers, and Rob H. Geleiderblom, "Mapping of H.264 decoding on a multiprocessor architecture," *Image and Video Communications and Processing 2003*, Proceedings of SPIE-IS&T Electronic Imaging, SPIE Vol. 5022, 2003
- [12] IBM workshop Cell Broadband Engine-An Introduction Sep. 2006
- [13] Fletcher, R.,*Practical Methods of Optimization*, Vol. 2:Constrained Optimization, John Wiley and Sons, Chichester, 1981
- [14] ISO/IEC JTC 1/SC 29/WG 11 N8244 "Joint Multi-view Video Model (JMVM) 3.0", Jan. 2007
- [15] Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG JVT-V209 "Joint Draft 2.0 on Multi-view Video Coding", Jan. 2007

V. CONCLUSION

Because of the parallelizability of MVC, CMP meets MVC's high performance requirement well. The performance of tasks distributing scheme referring to Selections of Optimal Theories & Methods is 8 times better than the serial algorithm. If the algorithm has parallel ability, large data volume and computing complexity like MVC, CMP architecture can reduce running time effectively. The combination of MVC and CMP is an appropriate solution.

The result can be improved. It will be enhanced more, if considering data I/O and storage, for example, by putting pieces of data which are processed in parallel in different storage units. We also can change frames processing order optimally, using SPUs more adequately.

VI. ACKNOWLEDGEMENT

The authors gracefully thank all the anonymous reviewers for their comments. This work is supported by the National Natural Science Foundation of China under Grant