A New Architecture of Distributed Video Coding For Real Time Systems

Peng Wang, Xiaodong Liu
Broadband Network and Multimedia Research Center
Graduate School at Shenzhen, Tsinghua University, China
E-mail:p-w06@mails.tsinghua.edu.cn,liuxd@sz.tsinghua.edu.cn

***Abstract*-Most of the reported Distributed Video Coding (DVC) schemes have a high time-delay in decoder which hinders its practical application in real-time systems. In this paper, we propose a novel DVC architecture based on MPI cluster in a practical scene, which includes three major approaches to accelerate decoder: an efficient parallel algorithm for generation of the Side Information (SI), a parallel method for LDPC iterative decoding process and a rate control approach based on machine learning for reducing feedbacks from decoder to encoder. The experimental results on cluster demonstrate significant savings in time with a little loss of video quality.**

I. INTRODUCTION

It is well known that the convetional video compression such as the ISO MPEG schemes or the ITU-T recommendations H.263 and H.264 has been driven predominately by the broadcasting and steaming video-on-demand applications and requires much more computation for encoding than for decoding. However, emerging applications such as security surveillance and mobile camera phones demand a low-power and low-complexity encoder. Motivated by these emerging applications, DVC schemes were developed in recent years.

The theoretical bases for distributed video coding are Slepian-Wolf [1] theorem and Wyner-Ziv [2] theorem in Networks Information Theory. These Theorems indicate that an intraframe encoder-interframe decoder system can come closer to the efficiency of interframe codec system such as H.26x schemes. Bernd Girod's group at Stanford and Kannan Ramchandran's group at Berkeley lead mainly the research on practical DVC solution [3][4]. In addition, DISCOVER [5], a European project funded under the European Commission IST FP6 programme, also works at the theoretical developments and practical schemes. Zixiang Xiong's group at TAMU [6] proposed layered Wyner-Ziv video coding framework for noisy channel.

Above mentioned DVC architectures face a common problem: high decoding complexity, which restrains them from being used in real-time video application such as video camera networks for surveillance. The complexity arises mainly from two factors: one is iterative LDPC (or Turbo) decoding process with a feedback channel, and the other is motion estimation procedure in the Side Information (SI) generation. In order to obtain a solution which is more suitable for practical applications, new ideas have been proposed to amend or to optimize the structure of decoder. Martinez *et al* presented a feedback free DVC architecture using machine learning approach [7], Weerakkody *et al* gave a unidirectional DVC architecture [8]. However, SI generation plays a key role in determining the performance of the codec and the reconstructed video quality is also sensitive to the side information. It is common that reduced cost on a motion search for a faster generation may cause a sharp decrease in PSNR. Besides, abundant channel decoding iterations guarantee decoding bits' accuracy, which is also critical to the video quality. For these reasons, instead of reducing some computing steps, we are inclined to adopt parallel approaches to achieve a real-time decoder with complete computation. With CPU chips becoming cheaper, multicore PC clusters are finding their broader applications. In addition, we also adopt a rate control method improved on the approach of [7], but our method has better PSNR performance since we extract more accurate features to generate the decision tree.

The paper is organized as follows. First, a new DVC architecture are introduced in section 2. Then, the proposed parallel algorithms for SI generation and LDPC iterative decoding process are discussed in section 3, the improved rate control method is also described in this section. Finally, section 4 concludes the paper.

II. ARCHITECTURE

A novel DVC architecture is designed based on MPI cluster which is more suitable for real-time applications (See Figure 1). At the encoders, one video sequence is divided into key frames and Wyner-Ziv (WZ) frames. Key frames are conventionally encoded and decoded by H264/AVC Intra coding approach. WZ frames are firstly transformed with DCT and quantized in a Transform Domain based scheme, or are directly quantized in a Pixel Domain based scheme, and then the bit planes of transform coefficients or pixels are encoded by using LDPC. At the decoder, the conventionally decoded previous and next key frames are inputted into the module of scheduling for SI generation, and then the side information approximating the WZ frames is generated by using our parallel motion searching algorithm. After that, the SI is used in a parallel LDPC decoder, along with WZ parity bits to reconstruct the decoded WZ frames. The proposed scheme does not need as many feedbacks from the decoder to encoder as traditional schemes and uses a decision tree similar to [8] for the estimation of the range of required bit-rate. In the case of the number of encoders is more than one, a scheduling module is taken to distribute the bit streams from encoders to different modules of the decoder for optimizing the computation of the cluster.

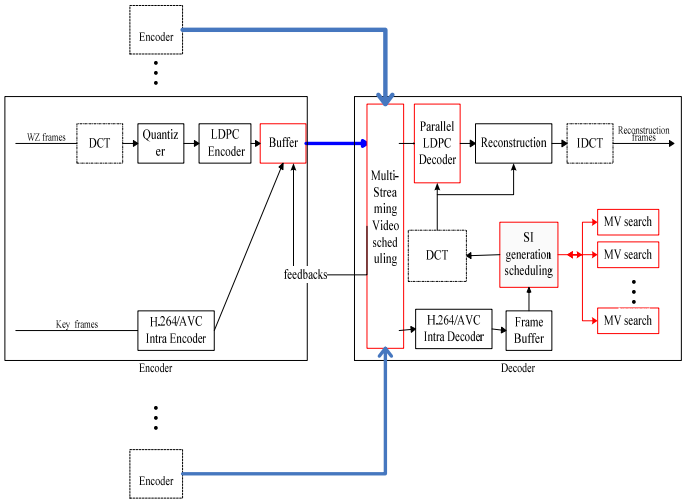

Fig. 1 A new DVC architecture based on cluster

III. ALGORITHMS

A. *Proposed parallel SI generating algorithm*

The serial counterpart of our parallel SI generating scheme used a symmetric Motion Vectors (MV) interpolating method presented by Aaron *et al* [9]. In our algorithm, we use a Group of Pictures (GOP) size of 2. Given a WZ frame WZ_{2i+1}, where $i = 0,1,2...$ is the GOP index. The adjacent key frames of the WZ frame are K_{2i} and K_{2i+2}. The frame WZ_{2i+1} was firstly divided into a series of blocks, and the size of each block is $m \times n$ pixels, where $m \times n$ can be 4×4, 8×8, 16×16, and 4×8 *etc*. For each block in frame WZ_{2i+1}, an important assumption is that the motion vector of the block from frame $2i$ to $2i+1$ is the same as the motion vector from frame $2i+1$ to $2i+2$. Based on this assumption, a searching area is defined for each block in K_{2i} and K_{2i+2}. The shape of the searching area is usually a rectangle which centered at the block having same index with the block in WZ_{2i+1}.Then a block matching process will find the best symmetric MV for this WZ block, which is approximated with the average of the best matching blocks in key frames. If the block size is too small and the searching area is too big, the searching process will be slow. On the contrary, if the block size is too big and the searching area is too small, we will have an inaccurate MV which always results in coarse side information even though the block searching is fast. Therefore, a suitable choice about the size of block and searching area should be made to achieve a trade-off between cost and performance.

In order to design an efficient parallel version of symmetric MV interpolation, one of the fundamental problems that we need to solve is to split the computing tasks of motion estimation into a set of sub-tasks for concurrent execution. Since the motion estimation of each block is independent of those of the others, we take the motion estimation of one block as the minimal Virtual Parallel Unit (VPU). Several VPUs are mapped onto a physical processing element. And those VPUs running on one physical unit are called a "task" and the size of a task, namely the number of VPUs that the task contains, is called granularity, which is critical to running performance. Besides, to achieve better load balancing, our scheme adopts a dynamic distribution of tasks instead of static predistribution.

Before introducing the parallel SI generating algorithm, we first denote N blocks with B_i $(i = 1,2...N)$ and P processors with P_i $(i = 1,2...P)$. The processors are classified into two categories: scheduling nodes and computing nodes. Here we assign P_0 as the scheduling node (see Fig.2), whose functions include: (1) Managing the task pool. (2) Initially distributing the tasks to the computing nodes. (3) Responding to the request of data exchange from computing nodes, receiving their estimated MVs, and sending new tasks and data to them. (4) Generating SI through interpolation after collecting MVs of all blocks. At the same time, P_i $(i \neq 0)$ is designated as the computing node, which implements motion estimation of the blocks.

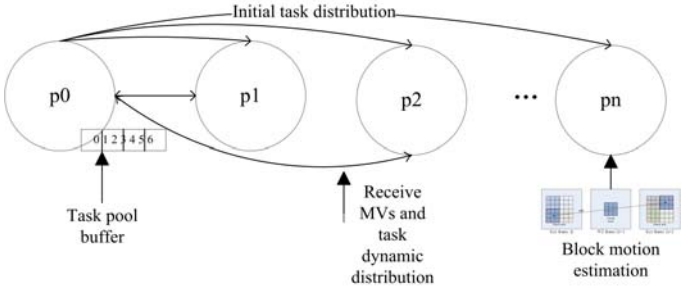

Fig. 2 An illustration of parallel SI generation

B. *Proposed parallel LDPC iterative decoding approach*

Our parallel LDPC iterative decoding method is based on the algorithm presented by Varodayan in [10]. Denote *OFD* as one frame decoding process, *OLD* as One LDPCA decoding process of a bit stream, *OBP* as One belief propagation process of a bit stream, *OBPI* as One iteration of the belief propagation. The relationships of these events can be described as one *Event tree*, which is shown in Fig.3, every parent node (parent event) consists of certain number of child nodes (child events).

Assume that the length of each block inputted into LDPC decoder is equal, and then the time of each *OBPI* is constant, so event *OBPI* can be treated as the minimal Virtual Parallel Unit. For reducing the cost of communication among nodes, several VPU should be mapped into one unit for computation. This should be designed carefully, so we take a hybrid strategy that combined static task distribution and the dynamic task distribution. We first make a predistribution of tasks in OLD level, and then dynamically adjust tasks on each node in OBPI level. If some processors are busy and other processors are idle, a portion of tasks in busy nodes are transited to idle nodes. Unlike above mentioned SI generation algorithm where only one schedule node exists, each node participates scheduling in

parallel LDPC decoding process.

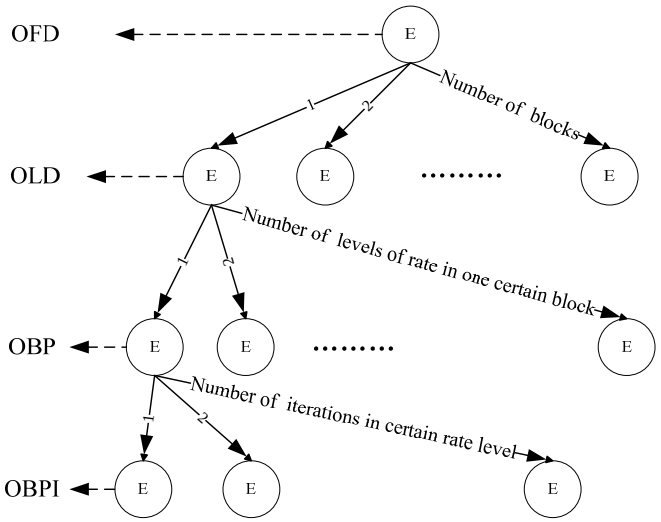

Fig.3 event tree in parallel LDPC decoding method

C. Rate control approach based on decision tree

Methods based on machine leaning to reduce even cancel feedbacks are first proposed by Martinez *et al* in [7], but two drawbacks exist in their approach: one is that the features such as mean and variance of the decoding block in their method are not accurate in the stage of training, with a weak classification performance in our experiment, the other is that a block with small size (4×4 in their paper) promises their good classification performance, but it is unpractical to take blocks with such a short length in Turbo/LDPC decoding, since it is well known that in order to explore the statistical relationship of bits in a stream, Turbo/LDPC module usually takes a block with long length, which usually consists of hundreds or thousands of bits.

To solve the problems mentioned above, we improved the approach in [7] by extracting more accurate features: hamming distance of each block and its sub-block between WZ frame and its corresponding coarse SI in the encoder. These features have directly relationship with the level of rate of a block in the LDPCA decoding process. Another benefit by adopting hamming distance is that a block with big size can be taken without weakening classification performance in the decision tree model. The relationship between the hamming distance and the level of rate can be seen in Fig.4, test sequence is the first 101 WZ frames of Foreman QCIF.GOP size is 2.

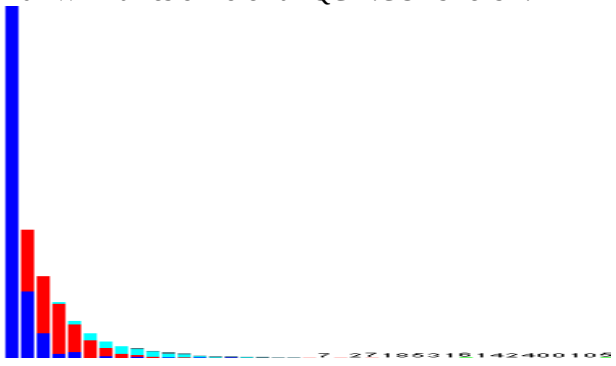

Fig.4 rate- hamming distance curve

In the figure 4, x-axis presents the hamming distance, y-axis presents the number of blocks of a video sequence in some value of hamming distance, different color represents different level of rate. It can be seen that with the change of hamming distance, the levels of rate vary quickly, which indicates a good classification performance. As in [7], software WEAK is used to train a decision tree.

IV. CONCLUSION

In this paper, a new DVC architecture with a decoder based on MPI cluster is proposed. This scheme contains three contributions: a novel parallel SI generation algorithm, a new parallel LDPC iterative decoding method, and an improved rate control approach based on decision tree in the encoder. This architecture is suitable for real-time applications based on distributed video coding such as video surveillance.

REFERENCES

[1] J. D. Slepian and J. K. Wolf, “Noiseless coding of correlated information sources,” IEEE Trans. Inf. Theory, vol. IT-19, pp. 471–480,Jul. 1973.

[2] A. D. Wyner, “Recent results in the Shannon theory,” IEEE Trans. Inf. Theory, vol. IT-20, no. 1, pp. 2–10, Jan. 1974.

[3] B. Girod, A. Aaron., et al. “Distributed video coding”. Proceedings of the IEEE Vol. 93, No. 1, pp:71–83 Jan.2005.

[4] S. S. Pradhan and K. Ramchandran, “Distributed source coding using syndromes (DISCUS): design and construction,” in Proc.IEEE Data Compression Conf., 1999, pp. 158–167.

[5] X. Artigas, J. Ascenso, M. Dalai, S. Klomp, D. Kubasov, M.Ouaret “The DISCOVER codec: Architecture, techniques and evaluation” Picture Coding Symposium 2007, Lisbon,Portugal, 2007.

[6] Xu Qian, XIONG Zi-xiang, “Layered Wynr-Ziv video coding” Proc of video Coding and Image Processing. 2004.

[7] Martinez, J.L. Fernandez-Escribano, G.Kalva, H.Weerakkody, W.A.R.J. Fernando, W.A.C.Garrido, A. “feedback free DVC architecture using machine learning”, Image processing 2008

[8] W.A.R.J.Weerakkody, W.A.C. Fernando, A.B.B. Adikari,“Unidirectional Distributed Video Coding for low cost video coding”, IEEE Transactions on Consumer Electronics, Vol. 53, No. 2, MAY 2007

[9] A. Aaron, R. Zhang, and B. Girod, “Wyner–Ziv coding of motion video,” presented at the Asilomar Conf. Signals and Systems, Pacific Grove, CA, 2002.

[10] D. Varodayan, A. Aaron, and B. Girod, “Rate-Adaptive Distributed Source Coding using Low-Density Parity-Check Codes”, 39th Asilomar Conference on Signals, Systems and Computers, Paci c Grove, CA, USA, Oct./Nov. 2005.