Fast Fine Granularity Scalability Decoding Scheme for Low-Delay Scalable Video Coding Applications

Nae-Ri PARK†, Joo-Hee MOON†, Nonmembers, and Jong-Ki HAN†, Member

SUMMARY The Fine Grain Scalability (FGS) technique used in SVC codec encodes and decodes the quantization error of QBL (quality base layer) along the cyclic scanning path. The FGS technique provides the scalability property to the compressed bit stream. However, the cyclic scanning procedure of FGS method may require a huge computing time. In this paper, we propose a fast FGS decoding scheme, which has a lower decoding complexity without sacrificing image quality.

key words: SVC (scalable video coding), FGS (fine granularity scalability), video coding

1. Introduction

The objective of SVC codec for internet streaming video is to optimize the image quality over a bit rate range instead of at a fixed bit rate. The enhancement bit stream generated from SVC encoder can be partially decoded by truncating the received stream at variable bit rates. The enhanced data is encoded by Fine Grain Scalability (FGS) technique along the cyclic scanning path. The FGS technique is one of schemes which enable the SVC codec to provide scalability property [1]–[4]. However, the Fine Grain Scalability module requires a huge computing time since the computational complexity of the cyclic scanning procedure is very high [1], [4].

The cyclic scan runs over 4×4 transformed blocks with a zig-zag pattern [2], [3]. Figure 1 illustrates the scan procedure of cyclic encoding and decoding when vector size is equal to 2 [5], [6]. In FGS encoding and decoding, some scans are for low frequency data, and others encode the high frequency data, since the cyclic entropy encoder is applied to the transformed data from low frequency to the high frequency components. Therefore, the image quality improvement resulting from a decoding scan is different from that affected from other scan. In this paper, we propose a fast FGS decoding scheme which puts a limitation on the number of cycles based on the number of all-zero blocks and the energy of QBL. The proposed scheme assigns many cycles to the frame whose QBL has large energy and whose the non-all-zero block ratio is large. As we can see from simulation results, the proposed algorithm reduces average 9.4% decoding time with very small degradation of PSNR.

2. The Conventional FGS Decoding Scheme

In FGS decoding procedure, the early and later scans are used for the low and high frequency components, respectively. If all blocks (whose size is 4×4) have zero coefficients, the number of total decoding cycles becomes 0. The cyclic scanning order has been modified to reduce the decoding time of FGS module [5], [6]. But, the complexity of FGS decoding module is still very high. Since most information of the transformed data is concentrated into DC or low frequency components, the low frequency components are decoded before the high frequency data along the zig-zag scan order as in Fig. 1 [5], [6]. The early decoding cycles provide much more effects on the quality of the reconstructed image than the later cycles, since the lower frequency region has more important data than the higher frequency region. No scanning is applied for a block which does not have any non-zero coefficient.

3. The Proposed FGS Decoding Scheme

We propose a fast FGS decoding scheme which can be applied to the low-delay SVC applications. Since the complexity of the FGS decoding module occupies about 60% of total complexity of the entire SVC decoder, the proposed algorithm enables the SVC decoder to be used for real time applications.

The proposed algorithm limits the number of cycles in FGS decoding, where many cycles are executed for a frame having many non-all-zero blocks and little cycles are used for that having many all-zero blocks. On other hand, the energy of QBL is considered to control the number of cycles. Many and little cycles are applied for frames whose QBL have large or small energy, respectively. The energies...
(E(i) and F(i)) of pixels in the QBL and FGS Layer of the ith frame are defined as

\[
E(i) = \sum_{(a,b) \in \text{QBL}} |p(a,b)|^2 \\
F(i) = \sum_{(a,b) \in \text{FGS Layer}} |f(a,b)|^2
\]  

(1)
(2)

where \(p(a,b)\) and \(f(a,b)\) are pixel values at the location \((a,b)\) in the QBL and FGS Layer of a frame, respectively.

The relation between \(E(i)\) and \(F(i)\) is shown in Fig. 2 which has been obtained by testing the decoder with "Foreman" and "Crew" sequences. As we can see in Fig. 2, when the QBL has high signal energy, there is much information in the FGS layer, i.e., many cycles are needed to decode the FGS layer. Thus, in the proposed algorithm, the number of the cycles is set to a value which is proportional to the energy of the QBL.

The proposed cycle limitation algorithm is represented in Fig. 3, where the ratio between numbers of non-all-zero blocks in the current decoded and the first frame is calculated to consider how many blocks are non-all-zero blocks in the decoded frame. The non-all-zero block ratio is represented as

\[
R_{NZ}(i) = \frac{\text{Num\_Blocks}(i) - \text{Num\_AllZeroBlocks}(i)}{\text{Num\_Blocks}(1) - \text{Num\_AllZeroBlocks}(1)}
\]

(3)

![Fig. 2 The relation between energies of QBL and FGS Layer.](image)

![Fig. 3 The flowchart of the proposed scheme.](image)
where $\text{Num. Blocks}(i)$ and $\text{Num. All Zero Blocks}(i)$ are the numbers of total blocks and all-zero blocks in the $ith$ frame, respectively. If the non-all-zero block ratio of the frame is large, the proposed decoding algorithm sets the number of cycles as a large value. Besides the non-all-zero-block ratio, the energy level of the frame is also considered. The relative energy level is calculated as a ratio between energies of the first and the current frame.

$$R_E(i) = \frac{E(i)}{E(1)}$$

If the decoded frame is a reference frame for other frames, we set the final ratio $R(i)$ as a maximum value between $R_{NZ}(i)$ and $R_E(i)$ to decode the frame with high quality. For other case, a minimum value is selected to restrict the decoding cycles. At the final step, the number of decoding cycles is set to $R(i) \times C(1)$ which is proportional to $C(1)$, the number of decoding cycles in the first frame.

4. Simulation Results

Simulations using “Foreman” and “Crew” sequences are performed to evaluate the proposed algorithm. We apply the proposed FGS decoding algorithm to the JSVM_6.5 reference software. In the simulations, the size of GOP is set to 16, the number of layers is 2 (QCIF and CIF). Closed loop is used to reconstruct the L and H frames. The condition of FRext is off. Inter-layer prediction is set as “adaptive”, decoding loop is implemented as a single loop.

Since the PSNR and the decoding time resulted from the proposed SVC decoder are affected from $C(1)$ (the number of cycles for the first frame), the setting $C(1)$ is important for the performance of the SVC decoder. To provide a criterion in selecting $C(1)$, we show the relationship between the saved decoding time and the decoding cycle number in Table 1, where the data is resulted by applying various training data such as “Foreman” and “Crew” sequences to the JSVM_6.5. Before we use our FGS decoding algorithm, we set the value of $C(1)$ which is related to the restricted decoding time.

The performances of the various FGS decoding schemes are compared in Table 2, where “Full FGS decoding” is a reference SW which decodes the data of the QBL and all data of one FGS layer. “JSVM_6.5(QBL)” does not use FGS data at all. “Fixed Cycles (5 cycles)” implies that the system executes the QBL decoding and only five FGS decoding cycles for all frames without considering the property of the frame. From the simulation results shown in Table 2, it can be observed that the better results are achieved by the proposed algorithm (with $C(1) = 10$) than “Fixed Cycles” and “JSVM_6.5 with QBL only”. In the comparison with JSVM_6.5 with full FGS decoding, the proposed algorithm reduces average 9.4% decoding time with very small degradation of PSNR. This is due to the fact that the proposed algorithm assigns the numbers of decoding cycles efficiently.

5. Conclusions

We propose a fast FGS decoding scheme which puts a limitation on the number of cycles based on the number of non-all-zero blocks and the energy of quality base layer. Since the proposed algorithm assigns the FGS cycle number to each frame efficiently, the decoding computation time is reduced by average 9.4% without any significant loss of image quality.

Acknowledgment

This work is supported by a grant of the Seoul R&B Program (10557).
References


