WHAT ARE GOOD CGS/MGS CONFIGURATIONS FOR H.264 QUALITY SCALABLE CODING?

Shih-Hsuan Yang and Wei-Lune Tang
Department of Computer Science and Information Engineering, National Taipei University of Technology
1 Sec. 3, Chung-Hsiao E. Rd., Taipei, Taiwan

Keywords: Scalable video coding, Rate-distortion performance, Coding complexity, H.264/AVC.

Abstract: Scalable video coding (SVC) encodes image sequences into a single bit stream that can be adapted to various network and terminal capabilities. The H.264/AVC standard includes three kinds of video scalability, spatial scalability, temporal scalability, and quality scalability. Among them, quality scalability refers to image sequences of the same spatio-temporal resolution but with different fidelity levels. Two options of quality scalability are adopted in H.264/AVC, namely CGS (coarse-grain quality scalable coding) and MGS (medium-grain quality scalability), and they may be used in combinations. A refinement layer in CGS is obtained by re-quantizing the (residual) texture signal with a smaller quantization step size (QP). Using the CGS alone, however, may incur notable PSNR penalty and high encoding complexity if numerous rate points are required. MGS partitions the transform coefficients of a CGS layer into several MGS sub-layers and distributes them in different NAL units. The use of MGS may increase the adaptation flexibility, improve the coding efficiency, and reduce the coding complexity. In this paper, we investigate the CGS/MGS configurations that lead to good performance. From extensive experiments using the JSVM (Joint Scalable Video Model), however, we find that MGS should be carefully employed. Although MGS always reduces the encoding complexity as compared to using CGS alone, its rate-distortion is unstable. While MGS typically provides better or comparable rate-distortion performance for the cases with eight rate points or more, some configurations may cause an unexpected PSNR drop with an increased bit rate. This anomaly is currently under investigation.

1 INTRODUCTION

Video is used in diversified situations. The same video content may be delivered in different and variable transmission conditions (such as bandwidth), rendered in various terminal devices (with different resolution and computational capability), and served for different needs. Adaptation of the same video content to every specific purpose is awkward and inefficient. Scalable video coding (SVC), which allows once-encoded content to be utilized in flexible ways, is a remedy for using video in the heterogeneous environments (Ohm, 2005).

Video scalability refers to the capability of reconstructing lower-quality video from partial bit streams. An SVC-coded signal is encoded once at the highest quality (resolution, frame rate) with appropriate packetization, and then can be decoded from partial streams for a specific rate or quality or complexity requirement. There are three categories of scalability in video: spatial (resolution), temporal (frame rate), and quality (fidelity). The major expenses of SVC compared to state-of-the-art non-scalable single-layer video coding are the gap in compression efficiency and increased encoder and decoder complexity.

The H.264 standard, also known as MPEG-4 AVC (Advanced Video Coding) (ITU-T Rec. H.264, 2009), has been dominating the emerging video applications including digital TV, mobile video, video streaming, and Blu-ray discs. The wide adoption and versatility of H.264/AVC leads to the inclusion of scalability tools in its latest extension (Schwarz and Marpe, 2007). There are two options for H.264 quality scalability, CGS (coarse-grain quality scalable coding) and MGS (medium-grain quality scalability). For CGS, a refinement of texture information is achieved by re-quantizing the (residual) texture signal with a smaller quantization step size (QP) relative to that used for the preceding

Yang S. and Tang W..
WHAT ARE GOOD CGS/MGS CONFIGURATIONS FOR H.264 QUALITY SCALABLE CODING?.
DOI: 10.5220/0003608201040109
In *Proceedings of the International Conference on Signal Processing and Multimedia Applications* (SIGMAP-2011), pages 104-109
ISBN: 978-989-8425-72-0

CGS layer. Inter-layer prediction may be employed to increase compression efficiency of CGS. However, the number of available bit rates is restricted to the number of selected QPs (CGS layers) and more layers generally imply worse coding efficiency. To increase the flexibility of bit stream adaptation and to improve the coding efficiency, MGS additionally provides the capability to distribute the CGS enhancement layer transform coefficients into more layers. Grouping information of the transform coefficients is signaled in the slice headers, and thus, a CGS layer that corresponds to a certain QP can be partitioned into several MGS layers and separately packetized. Pulipaka et al (2010) conducted some statistical analyses of SVC, including the rate distortion and rate variability distortion performances. Görkemli et al (2010) compared MGS fragmentation configurations of SVC, including the slice mode and extraction methods, for their rate-distortion performance.

In this paper, we test various CGS/MGS options for H.264 SVC using the official reference software JSVM (Joint Scalable Video Model) (JSVM Software Manual, 2010/2011). Throughout the comprehensive experiments, unusual rate-distortion behavior for some configurations of SVC options was discovered. It is generally believed that an additional quality layer (more received bits) should always improve the quality for SVC. However, we find that adding an MGS sub-layer in some cases may conversely decrease the PSNR. We thus conduct more tests to explore this anomaly. The rest of this paper is organized as follows. In Section 2, we briefly review the H.264 SVC techniques, particularly in details for CGS and MGS. Experiments on H.264 quality scalability with various JSVM CGS/MGS configurations are given in Section 3, which also demonstrates the aforementioned oddity. Some discussion and future work are given in Section 4.

2 H.264 SCALABLE VIDEO CODING

H.264 includes two layers in structure: video coding layer (VCL) and network abstraction layer (NAL). Based on the core coding tools of the non-scalable H.264 specification, the SVC extension adds new syntax for scalability (ITU-T Rec. H.264, 2009). The representation of the video source with a particular spatio-temporal resolution and fidelity is referred to as an SVC layer. Each scalable layer is identified by a layer identifier. In JSVM, three classes of identifiers, *T*, *D*, and *Q*, are used to indicate the layers of temporal scalability, spatial scalability, and quality scalability, respectively. A constrained decoder can retrieve the necessary NAL units from an H.264 scalable bit stream to obtain a video of reduced frame rate, resolution, or fidelity. The first coding layer with identifier equal to 0 is called the base layer, which is coded in the same way as non-scalable H.264 image sequences. To increase coding efficiency, encoding the other enhancement layers may employ data of another layer with a smaller layer identifier.

Temporal scalability provides coded bit streams of different frame rates. The temporal scalability of H.264 SVC is typically structured in hierarchical B-pictures. In this case, each added temporal enhancement layer doubles the frame rate. These dyadic enhancement layer pictures are coded as B-pictures that use the nearest temporally available pictures as reference pictures. The set of pictures from one temporal base layer to the next is referred to as a group of pictures (GOP). It is found from experiments that the GOP size of 8 or 16 usually achieves the best rate-distortion performance (Schwarz and Marpe, 2007). Note that the GOP size also determines the total number of temporal layers (no. of temporal layers = ($\log_2$ GOPsize) + 1).

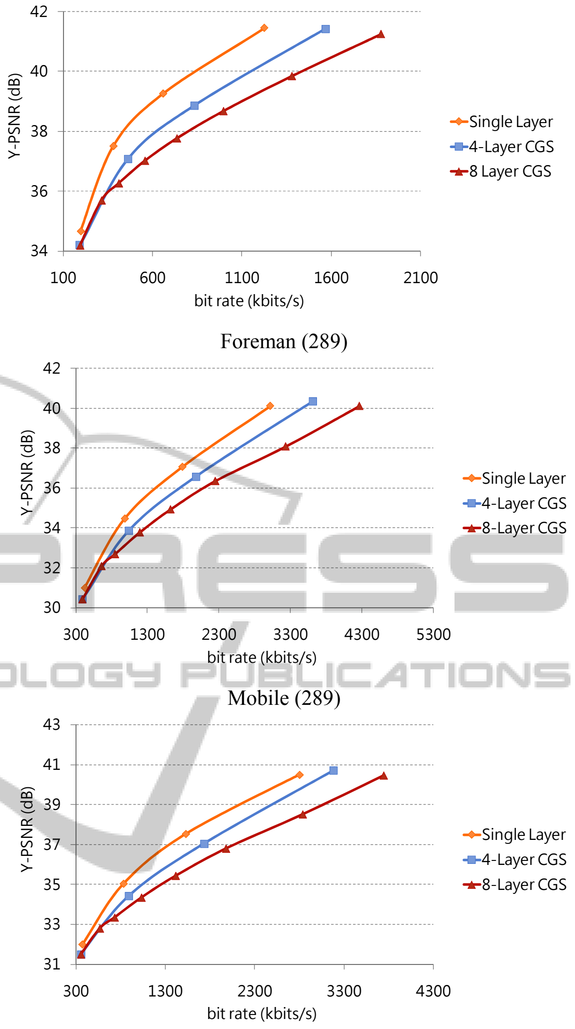
Each layer of H.264 spatial scalability corresponds to a specific spatial resolution. In addition to the basic coding tools of non-scalable H.264, each spatial enhancement layer may employ the so-called interlayer prediction, which employs the correlation from the lower layer (resolution). There are three prediction modes of inter-layer coding: inter-layer intra prediction, inter-layer motion prediction, and inter-layer residual prediction. Accordingly, the up-sampled reconstructed intra signal, the macroblock partitioning and the associated motion vectors, or the up-sampled residual derived from the colocated blocks in the reference layer, are used as prediction signals. The inter-layer prediction shall compete with the intra-layer temporal prediction for determining the best prediction mode.

Quality scalable layers, which are the main concern of this paper, have identical spatio-temporal resolution but different fidelity levels. H.264 offers two options for quality scalability, CGS (coarse-grain quality scalable coding) and MGS (medium-grain quality scalability). An enhancement layer of CGS is obtained by requantizing the (residual) texture signal with a smaller quantization step size (quantization parameter, QP). CGS incorporates the inter-layer prediction mechanisms very similar to those used in spatial scalability, but with the same picture sizes for the base and enhancement layers. Besides, the up-sampling operations and the inter-layer de-blocking for intra-coded reference layer macroblocks are omitted. Also, the inter-layer intra and inter-layer residual predictions are directly performed in the transform domain. SVC supports up to 8 CGS layers but the inter-layer prediction is constrained to at most three CGS layers including the required base layer. Usually, a significant difference in QP, which corresponds to largely deviated bit rates, is expected in order to achieve good RD performance (Schwarz and Marpe, 2007), (Pulipaka, Seeling, Reisslein and Karam, 2010). In Figure 1, 4-layer CGS and 8-layer CGS are compared with the non-scalable single-layer H.264 coding. More notable PSNR losses are observed for 8-layer CGS, as expected.

MGS is proposed in SVC to increase the adaptation flexibility, improve the coding efficiency, and reduce the coding complexity. A CGS layer that corresponds to a certain QP can be partitioned into several MGS sub-layers and distributed over different NAL units. An MGS sub-layer corresponds to a group of transform coefficients of 4×4 blocks in the zigzag order. The first and the last scan index for transform coefficients are signaled in the slice headers. Thus, the slice data (and the corresponding NAL units) may only include the indicated transform coefficients for a certain QP. The MGS sub-layers can more flexibly switch in any access unit in contrast that the CGS layers can only be changed in the next GOP. JSVM further limits the total number of rate points not exceeding 16, counting both the CGS layers and the MGS sub-layers. Note that at most 8 CGS layers are allowed and a large number of CGS layers may incur significant PSNR degradation and encoding complexity. Therefore, it may be preferable to incorporate MGS quality sub-layers inside some CGS layers if more rate points (say more than 4) are expected. However, some unusual rate-distortion performance is observed for some MGS configurations, as detailed in the next section.

3 OBSERVED ANOMALY IN H.264 QUALITY SCALABILITY

We comprehensively evaluate the rate-distortion performance and computational complexity for H.264 quality scalability, with focuses on



Foreman (289)

Mobile (289)

Tempete (257)

Figure 1: Comparisons of 4-layer and 8-layer CGS with the non-scalable single-layer coding. 4-layer CGS: QP = 36-28-24-20, 8-layer CGS: QP = 36-32-30-28-26-24-22-20. The single layer is individually coded for each QP (36-28-24-20) with non-scalable H.264 coding. The number after the sequence name indicates the number of total coded frames. Simulation is based on JSVM 9.19.9, and the results with JSVM 9.19.13 show little difference.

CGS/MGS configurations. Experiments were formerly conducted with JSVM 9.19.9 and later with JSVM 9.19.13, on nine test sequences shown in Figure 2. In JSVM, the primary encoding parameters are specified in the Main Configuration File (main.cfg), and the encoding parameters associated with each CGS layer are specified in the individual Layer Configuration File (layer*x*.cfg) where *x* denotes the dependency_id. A typical Main Configuration File and a typical Layer Configuration

File are shown in Table 1, where only the parameters important to our evaluations are listed.

The GOPsize is fixed to be 16. We thus set the number of frames to be encoded as a multiple of 16 plus 1, to its maximum. (The extra one is added for accomplishing the reference pictures of the hierarchical B structure.) Hence, 289 frames are used for image sequences of 300 frames. **`EncodeKeyPictures`** controls the drift of prediction. The value set to 1 means that pictures with MGS ($Q > 0$) refinement are coded as key pictures. Only minor variation is observed if we use another **`EncodeKeyPictures`** value and the global rate-distortion trend does not change. JSVM allows only three 3 CGS layers if we set **`CgsSnrRefinement`** = 0. Therefore, we set **`CgsSnrRefinement`** = 1 (MGS) along with appropriate LayerCfgs. It is also found that the value of **`CgsSnrRefinement`** will not change the coding results if no more than 3 CGS layers are used. Thus, 3-layer CGS (**`CgsSnrRefinement`** = 0) and 3-layer MGS (**`CgsSnrRefinement`** = 1) have almost identical rate-distortion results.

The parameter **`NumLayers`** specifies the total number of spatial/CGS layers. (Recall that CGS is regarded as a special case of spatial scalability.) In a Layer Configuration File that corresponds to a specific QP, the parameter **`MGSVectorMode`** specifies whether MGS is used, i.e., whether the transform coefficients of the CGS layer are written into several MGS quality layers according to **`MGSVectorX`**. The parameter **`MGSVectorX`** specifies the number of transform coefficients in the $\mathbf{X}^{th}$ MGS sub-layer, i.e., the $\mathbf{X}^{th}$ position of the vector in the zigzag order.

Table 1: Encoding parameters.

(a) main.cfg

Parameter	Value	Remarks
`FrameRate`	30.0	
`FramesToBeEncoded`	289	No. of frames
`GOPSize`	16	
`CgsSnrRefinement`	1	1: MGS; 0: CGS
`EncodeKeyPictures`	1	MGS
`MGSControl`	2	ME+MC with EL, closing prediction loop at lowest and highest rate point
`SearchMode`	4	FastSearch
`SearchRange`	32	In full pels
`NumLayers`	4	CGS layers
`LayerCfg`	layer0-3.cfg	Layer configuration file

(b) layer3.cfg

Parameter	Value	Meaning
`SourceWidth`	352	Input frame width
`SourceHeight`	288	Input frame height
`InterLayerPred`	2	Inter-layer Prediction (0: no, 1: yes, 2:adaptive)
`MGSVectorMode`	1	MGS vector usage selection
`MGSVector0`	4	Specifies 0th position of the vector
`MGSVector1`	4	Specifies 1st position of the vector
`MGSVector2`	8	Specifies 2nd position of the vector
`QP`	20	Quantization parameters

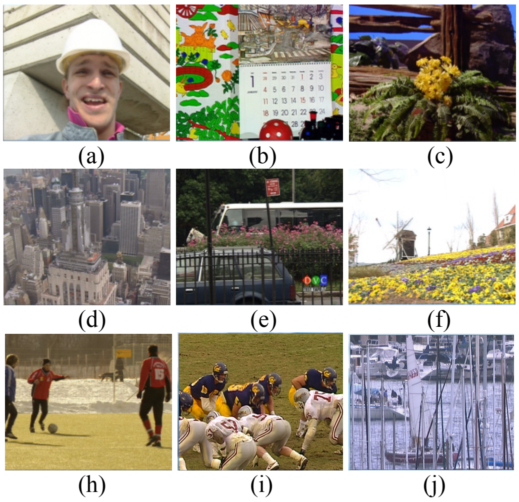
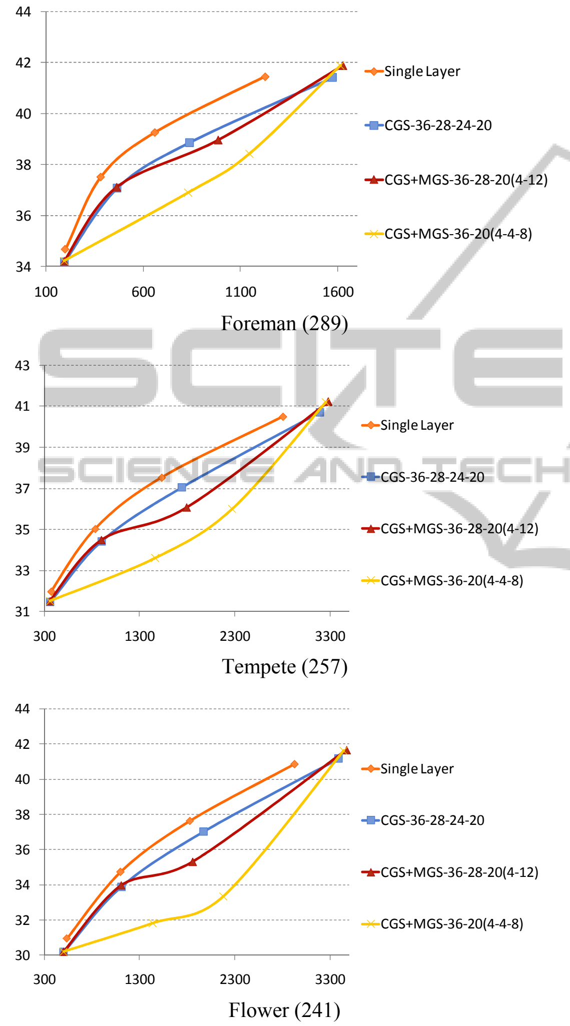

(a) (b) (c)
(d) (e) (f)
(h) (i) (j)

Figure 2: Test sequences, (a) Foreman, (b) Mobile, (c) Tempete, (d) City, (e) Bus, (f) Flower, (h) Soccer, (i) Football, (j) Harbour.

In the following, we present the simulation results for two cases, 4 rate points and 8 rate points. Due to the page limit, we primarily present the results for the three sequences Foreman, Tempete, and Flower. The results for 4 rate points are shown in Figure 3. Three SVC configurations are examined: (i) 4-layer CGS, QP = 36-28-24-20; (ii) 3-layer CGS, QP = 36-28-20(4-12); (iii) 2-layer CGS, QP = 36-20(4-4-8). A number in parentheses after a QP value denotes an **`MGSVector`**. Therefore, Configuration (ii), 36-28-20(4-12), indicates that 2 MGS sub-layers exist for the CGS layer with QP = 20 and these two sub-layers consist of 4 and 12 transform coefficients, respectively. As shown in Figure 3, the inserted MGS sub-layers degrade PSNR performance. However, incorporating MGS significantly reduces the encoding time as compared to using CGS alone, as shown in Figure 4. On the other hand, whether MGS is used has little effect on the decoding time. Although the results are shown only for three test sequences, the others generally exhibit similar behaviors.



Foreman (289)

Tempete (257)

Flower (241)

Figure 3: Rate-distortion results (4 rate points) with JSVM 9.19.13.

The rate-distortion results for 8 rate points are shown in Figure 5 (with JSVM 9.19.9) and Figure 6 (with JSVM 9.19.13). Three SVC configurations are examined: (i) 8-layer CGS, QP = 36-32-30-28-26-24-22-20; (ii) 4-layer CGS, QP = 36-32-26(4-4-8)-20(4-4-8); (iii) 3-layer CGS, QP = 36-28(4-4-8)-20(2-4-5-5). Adding MGS sub-layers outperforms the pure CGS layers in most rate points for most test sequences. The inefficiency of CGS attributes to the decreasing inter-layer correlation with dense QP settings. However, the first MGS sub-layer for QP = 20 may exhibit an unusual PSNR drop for some test sequences (Mobile, Tempete, Bus, Flower) with JSVM 9.19.9! A decreased PSNR is observed with more received bits, and then the rate-distortion plots gradually go back to their normal values. This anomaly becomes less significant in the newest JSVM 9.19.13 but is not fully resolved. We conduct some more tests on the Flower sequence, which yields the severest PSNR drop. It seems that the drop occurs at the first MGS sub-layer of the last CGS layer that has the smallest QP, and then the rate-distortion plots will gradually return to their normal positions. The encoding time comparison is shown in Figure 7, which confirms the time savings of MGS.

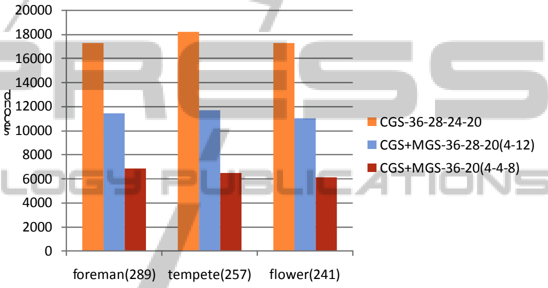
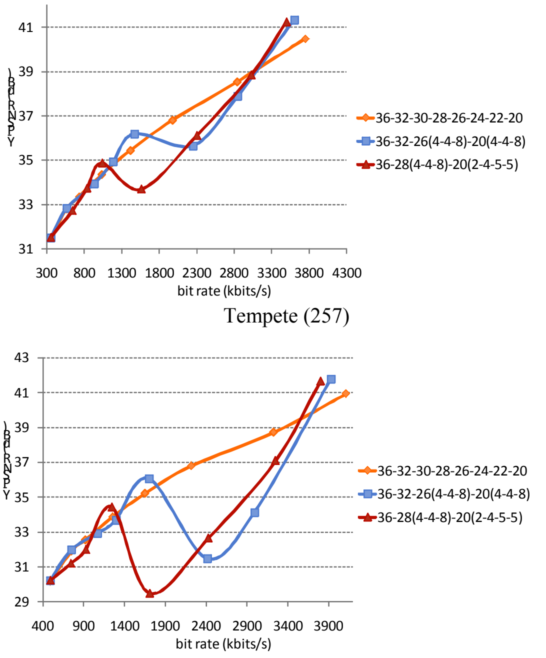


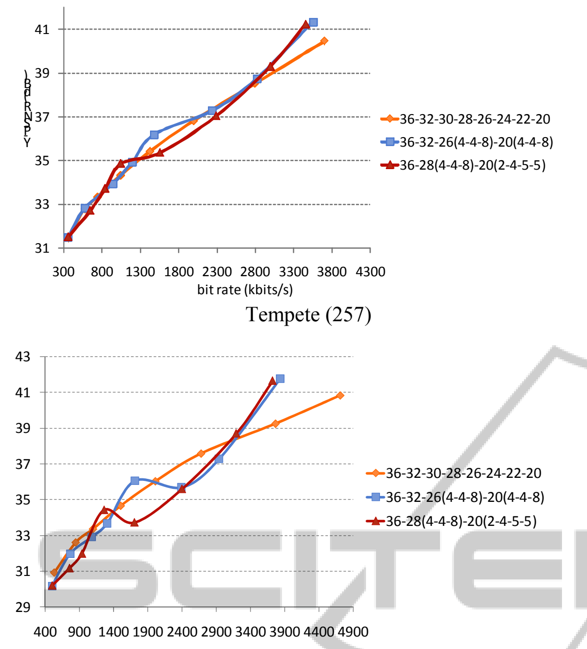
Figure 4: Encoding time comparison (4 rate points) with JSVM 9.19.13.



Tempete (257)

Flower (241)

Figure 5: Rate-distortion results (8 rate points) with JSVM 9.19.9.



Tempete (257)

Flower (241)

Figure 6: Rate-distortion results (8 rate points) with JSVM 9.19.13.

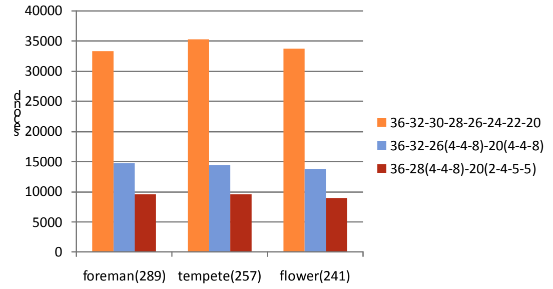


Figure 7: Encoding time comparison (8 rate points) with JSVM 9.19.13.

4 DISCUSSION AND FUTURE WORK

The effects of CGS/MGS configurations of H.264 SVC are investigated in this paper. For four or fewer rate points, CGS coding alone gives better coding performance despite of its high encoding complexity. For more rate points, adding MGS sub-layers to existing CGS layers may give better or worse rate-distortion performance as compared to using CGS alone. It is observed that some CGS/MGS configurations may cause an unexpected PSNR drop with an increased bit rate. The drop occurs at the first MGS sub-layer of the last CGS layer that has the smallest QP. Although this phenomenon is less significant in the latest version of JSVM, the problem is not fully resolved. As a consequence, one may be hesitant to use MGS due to its un-stability. The reasons behind this anomaly are under study. By investigating the source code and simulation results, our current finding and conjecture are towards the residual coding and the inter-layer prediction of blocks smaller than 8x8.

ACKNOWLEDGEMENTS

This work is supported in part by the National Science Council, R.O.China, under Grant NSC 99-2220-E-027-002.

REFERENCES

Ohm, J.-R., 2005. ‘Advances in scalable video coding’, *Proceedings of the IEEE*, vol. 93, no. 1, Jan, pp. 42-54.

ITU-T Rec. H.264, 2009. (MPEG-4 AVC), Fifth Edition, May (including SVC and MVC extensions).

Schwarz, H. and Marpe, D., 2007. ‘Overview of the scalable video coding extension of the H.264/AVC standard’, *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, Sep, pp. 1103-1102.

Pulipaka, A., Seeling, P., Reisslein, M and Karam, L.J., 2010. ‘Overview and traffic characterization of coarse-grain quality scalable (CGS) H.264 SVC encoded video’, *Consumer Communications and Networking Conference* (CCNC), Jan, pp.1-5.

Görkemli, B., Şadi, Y. and Tekalp, A.M., 2010. ‘Effects of MGS fragmentation, slice mode and extraction strategies on the performance of SVC with medium-grained scalability’, *IEEE International Conference Image Processing* (ICIP), Sep, pp.4201-4204.

JSVM Software Manual, 2010/2011. Version: 9.19.9 (CVS tag: JSVM_9_19_9), January 16th, 2010. JSVM Software Manual, Version: 9.19.13 (CVS tag: JSVM_9_19_13), May 4, 2011.