Dependent Quantization for Scalable Video Coding

*Danu Pranantha *김문철 **함상진 **이근식 **박근수

*한국정보통신대학교 (ICU)

**한국방송 (KBS)

Dependent Quantization for Scalable Video Coding

*Danu Pranantha *Kim, Munchurl **Hahm, Sangjin **Lee, Keunsik **Park, Keunsoo

*Information and Communications University (ICU)

**Korean Broadcasting System (KBS)

Abstract

Quantization in video coding plays an important role in controlling the bit-rate of compressed video bit-streams. It has been used as an important control means to adjust the amount of bit-streams to allowed bandwidth of delivery networks and storage. Due to the dependent nature of video coding, dependent quantization has been proposed and applied for MPEG-2 video coding to better maintain the quality of reconstructed frame for given constraints of target bit-rate. Since Scalable Video Coding (SVC) being currently standardized exhibits highly dependent coding nature not only between frames but also lower and higher scalability layers where the dependent quantization can be effectively applied, in this paper, we propose a dependent quantization scheme for SVC and compare its performance in visual qualities and bit-rates with the current JSVM reference software for SVC. The proposed technique exploits the frame dependences within each GOP of SVC scalability layers to formulate dependent quantization. We utilize Lagrange optimization, which is widely accepted in R-D (rate-distortion) based optimization, and construct trellis graph to find the optimal cost path in the trellis by minimizing the R-D cost. The optimal cost path in the trellis graph is the optimal set of quantization parameters (QP) for frames within a GOP. In order to reduce the complexity, we employ pruning procedure using monotonicity property in the trellis optimization and cut the frame dependency into one GOP to decrease dependency depth. The optimal Lagrange multiplier that is used for SVC is equal to H.264/AVC which is also used in the mode prediction of the JSVM reference software. The experimental result shows that the dependent quantization outperforms the current JSVM reference software encoder which actually takes a linear increasing QP in temporal scalability layers. The superiority of the dependent quantization is achieved up to 1.25 dB increment in PSNR values and 20% bits saving for the enhancement layer of SVC.

1. Introduction

JVT (Joint Video Coding), a joint group of MPEG (Motion Picture Expert Group) and VCEG (Video Experts Group), is working on the standardization of SVC (Scalable Video Coding) which is an extension to H.264/AVC(Advanced Video Coding). The SVC standard aims at providing the technologies for flexible representation of its compressed bit-streams to make it possible to cope with various terminal sizes and a wide range of network bandwidth etc. Hence, the SVC constructs compressed bit-streams with respect to three scalabilities such as temporal, spatial, and quality/SNR scalabilities. In each scalability layer of SVC, the first scalabilitylayer is called the base layer, whereas all its higher scalability layers, called the enhancement layers, are built on top of the base layer.

As aforementioned, the SVC is intended to support the video consumption with the available resources in the network and the clients through bit-stream truncation and SNR scalability. Yet, we want to keep the quality of the video as best as possible with the least amount of bit as possible within each scalability layer so that we may have more efficient amount of bit-stream for storage and transmission. There are two applicable techniques that can be applied to achieve an efficient amount of bit-stream. Firstly, we can optimize the operational control of an SVC through mode selection e.g. INTRA, INTER16x16, etc and motion prediction based on the R-D (rate-distortion) profile of video sequences [4], [5], [6], [7]. By doing this, we are able to reduce error in the reconstructed frames which will decrease the amount of bit-stream produced by the encoder. Secondly, we can optimize the QP (quantization parameter) selection for each frame because the quantization process plays an important role in controlling the amount of compressed output bit-stream [2], [3]. In this paper, we are focusing on the QP selection optimization to control the amount of output bit-stream produced by the SVC encoder.

The current JSVM (Joint Scalable Video Model) reference software for SVC takes a predefined fixed quantization approach for whole macroblocks in a frame with a linearly increasing QP setin the temporal scalability to control the bit-streams output [1]. The fixed value is chosen to maintain good quality in the base layer which will be used to predict the enhancement layer in temporal scalability axis. However, we may exploit the fact that, in SVC a coding layer depends on its lower layers. That is, the quantization at a lower layer affects the visual quality on the reconstructed frames of its higher layers. Hence, by examining this property, we propose a bit-rate control mechanism for SVC using dependent quantization based on the R-D profile of the video sequence. The objective of dependent quantization is to determine an optimal set of QP for frames within each GOP in each scalability layer by minimizing the R-D cost given a constraint of target bit-rate. In this paper, we utilized Lagrangian cost function to measure the cost of the bit usage and the distortion of the reconstructed frames. We also populated a trellis graph whose nodes and branches represent the costs of choosing certain QP for each frame within a layer. Furthermore, we employed a pruning procedure using the monotonicity property [2] so that the computational complexity in gathering the costs profile is reduced. The minimum path that survives in the trellis graph represents the best QP set for whole frames within a GOP.

This paper is organized as follow: in Section 2, we briefly describe the fundamental concept of SVC and the current quantization approach in JSVM in Section 3, we describe an example of formulation of dependent quantization for SVC along with its minimization technique and Lagrange multiplier selection in Section 4, we present the experimental results for dependent quantization and finally, in Section 5 we conclude our paper.

2. Scalable Video Coding

2.1. Concept

The SVC standard defines a technique to encode a video sequence into several scalability layers with respect to three scalability axes: temporal, spatial, and quality or CGS (coarse grain scalability) scalability. Fig. 1 illustrates the scalability dependency between scalability layers and the techniques used in each scalability axis. Firstly, the temporal scalability in SVC is achieved by a hierarchical B-Picture structure to exploit the temporal dependency between frames in each GOP. Unlike the precedent standards such as MPEG-2 and MPEG-4 AVC, the SVC can use B-Pictures as reference frames for prediction. Fig. 2 shows an example of a hierarchical B-Picture structure using single frame reference in forward and backward directions indicated by the arrows.

Secondly, for spatial and CGS scalability, the inter-layer texture prediction in Fig. 1 means that we encode the difference between the reconstructed higher layer and its reconstructed lower layer, whereas the inter-layer residual prediction means that we encode the residual of different between the residual of higher layer and its residual of lower layer. In addition, the inter-layer motion prediction means the higher layer uses motion vector of its lower layer. Although the techniques adapted in spatial and CGS scalability is the same, they are differ in term of spatial resolution. Spatial scalability has different spatial resolution while CGS scalability has the same spatial resolution. By observing aforementioned techniques, we can derive the frame and layer dependency which will be used in formulating the dependent quantization. The formulation will be presented in Section 3.

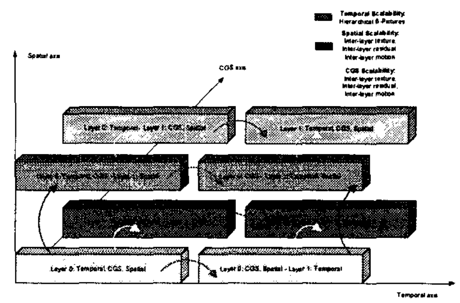

Fig. 1. Three scalability dependency in SVC.

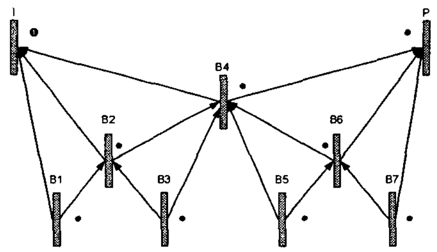

Fig. 2. An example of hierarchical B-Picture structure using single frame reference in a GOP.

2.2. Quantization in the JSVM Reference Software

The SVC standard defines a technique to encode a video sequence into several scalability layers with respect to three scalability axes: temporal, spatial, and quality or CGS (coarse grain scalability) scalability. Fig. 1 illustrates the scalability dependency between scalability layers and the techniques used in each scalability axis. Firstly, the temporal scalability in SVC is achieved by a hierarchical B-Picture structure to exploit the temporal dependency between frames in Basically, for each spatial or CGS layer within each GOP, QP has to be determined by high-level encoder control [1]. High-level encoder control can be done either by dependent quantization or other techniques. Nonetheless, JSVM reference software sets a configuration parameter called QP_0 to be fixed during the coding process throughout video sequence [1]. Suppose there are N numbers of temporal scalability level inside a spatial or CGS layer, the QP is adjusted for each hierarchy level $\ell = 0..N\text{-}1$ using Eq. (1) in JSVM reference software [1].

$$QP(\ell) = \min(51, \max(QP_0 + (\ell != 0 ? 2 : 0) \times 1.7 \times (N-1-\ell))) \quad (1)$$

Eq. (1) gives linearly increasing QP with the slope 1.7. By using linearly increasing QP, the JSVM reference software tries to preserve good quality in the lower layer which will be used to predict the higher layer in temporal axis, while the rate control is provided by SNR scalability. However, we should consider one of the interesting facts in SVC where one scalability layer depends on its lower layers. For example, the reconstruction of the second enhancement layer in temporal axis requires the preceding reconstruction of the first enhancement layer in temporal axis. By utilizing this dependency characteristic, we can achieve better decision of QP value set.

3. Dependent Quantization

3.1. Formulation

In this formulation sub-section, we will discuss about the frame dependency formulation for temporal scalability and spatial scalability in SVC. As an example, we will use single frame reference in both forward and backward motion prediction when defining the dependent quantization for SVC.

Table 1. Dependency table for single frame reference in temporal axis which refers to Fig.2

Temporal dependency ID	Dependency
1	I (independent)
2	I → P
3	I, P → B4
4	I, P, B4 → B2
5	I, P, B4 → B6
6	I, P, B4, B2 → B1
7	I, P, B4, B2 → B3
8	I, P, B4, B6 → B5
9	I, P, B4, B6 → B7

Firstly, we formulate the frame dependency in the temporal scalability within a GOP in the base spatial layer with four temporal scalability layers.

As aforementioned in Section 2 and illustrated in Fig. 2, we extend the layers in the temporal scalability by using hierarchical B-Pictures. Thus, we can define the temporal dependency between frames (indicated by the arrows) shown in Table 1 with its temporal dependency IDs. For the frame dependency in Table 1, we can formulate a constrained equation in Eq. (2), where D is distortion of a frame, R is bit-rate produced by a frame, R_{budget} is bit-rate constraint, and superscript "*t*" is "temporal scalability".

Eq. (2) can be simplified into Eq. (3) by utilizing the temporal dependency IDs in Table 1, where "*tid*" in Eq. (3) indicates temporal dependency ID.

$$\min \sum_{i=1}^{9} D_i^t(tid = i) \quad \textit{subject to} \qquad (3)$$
$$\sum_{i=1}^{9} R_i^t(tid = i) < R_{budget}$$

Table 2. Dependency table for the base layer and the first enhancement layer of spatial scalability.

Dependency ID for Base Layer of Spatial - First Enhancement Layer of Spatial (sid0 - sid1)	Base layer of Spatial Scalability Dependency	First EH layer of Spatial Scalability Dependency (dependent toward the base layer of spatial scalability in the left side)
1 - 1	I_{s0} (independent)	I_{s1}
2 - 2	$I_{s0} \rightarrow P_{s0}$	$I_{s1} \rightarrow P_{s1}$
3 - 3	I_{s0}, $P_{s0} \rightarrow B4_{s0}$	I_{s1}, $P_{s1} \rightarrow B4_{s1}$
4 - 4	I_{s0}, P_{s0}, $B4_{s0} \rightarrow B2_{s0}$	I_{s1}, P_{s1}, $B4_{s1} \rightarrow B2_{s1}$
5 - 5	I_{s0}, P_{s0}, $B4_{s0} \rightarrow B6_{s0}$	I_{s1}, P_{s1}, $B4_{s1} \rightarrow B6_{s1}$
6 - 6	I_{s0}, P_{s0}, $B4_{s0}$, $B2_{s0} \rightarrow B1_{s0}$	I_{s1}, P_{s1}, $B4_{s1}$, $B2_{s1} \rightarrow B1_{s1}$
7 - 7	I_{s0}, P_{s0}, $B4_{s0}$, $B2_{s0} \rightarrow B3_{s0}$	I_{s1}, P_{s1}, $B4_{s1}$, $B2_{s1} \rightarrow B3_{s1}$
8 - 8	I_{s0}, P_{s0}, $B4_{s0}$, $B6_{s0} \rightarrow B5_{s0}$	I_{s1}, P_{s1}, $B4_{s1}$, $B6_{s1} \rightarrow B5_{s1}$
9 - 9	I_{s0}, P_{s0}, $B4_{s0}$, $B6_{s0} \rightarrow B7_{s0}$	I_{s1}, P_{s1}, $B4_{s1}$, $B6_{s1} \rightarrow B7_{s1}$

Secondly, we have three techniques that are incorporated in the spatial scalability: inter-layer texture prediction, inter-layer residual prediction, and inter-layer motion prediction. An example shown in Table 2 is the dependency within a GOP in the base layer and the first enhancement layer (first EH) of spatial scalability with four temporal scalability layers. The first enhancement layer of spatial scalability is dependent toward the base layer of spatial scalability and each layer of spatial layer has its own temporal scalability.

By incorporating the spatial dependency IDs in Table 2, we can formulate the constrained equation in (4) where superscript "*sid0*" indicates the base layer of spatial scalability with its own temporal dependency, while the superscript "*sid1*" indicates the first enhancement layer of spatial scalability with its temporal dependency and its dependency toward the base layer of spatial scalability.

$$\min\left[\sum_{i=1}^{9} D_i^{s0}(sid0 = i) + \sum_{i=1}^{9} D_i^{s1}(sid0 = i, sid1 = i)\right]$$
$$\textit{subject to} \qquad (4)$$
$$\sum_{i=1}^{9} R_i^{s0}(sid0 = i) + \sum_{i=1}^{9} R_i^{s1}(sid0 = i, sid1 = i) < R_{budget}$$

3.2. Optimization Problem

In the sub-section above, we have two constrained equations in Eq. (3) and (4) which has to be solved. The next step is to consider the optimization technique for minimizing the distortion given bit-rate constraint in Eq. (3) and (4) above. We can solve the constrained equation in Eq. (3) and (4) by changing them using Lagrange multiplier λ, which is widely accepted in R-D based optimization [5], into unconstrained equations. The usage of Lagrange multiplier can be shown by changing the constrained equation in Eq. (4) above to unconstrained equation in Eq. (5) below, where J denotes the cost function formed by using Lagrange multiplier λ shown in Eq. (6).

$$\min\left[\sum_{i=1}^{9} J_i^{s0}(sid0 = i) + \sum_{i=1}^{9} J_i^{s1}(sid0 = i, sid1 = i)\right] \qquad (5)$$

$$J_i^{s0}(sid0 = i) = D_i^{s0}(sid0 = i) + \lambda \times R_i^{s0}(sid0 = i)$$
$$J_i^{s1}(sid0 = i, sid1 = i) = D_i^{s1}(sid0 = i, sid1 = i) + \lambda \times R_i^{s1}(sid0 = i, sid1 = i) \qquad (6)$$

The optimization problem of Eq. (5) can be easily solved by using trellis graph. Trellis graph technique tries to find the optimal path that gives the minimum cost within the trellis graph. Both of the nodes and the branches in the trellis graph represent the costs, while the optimal path represents the minimum cost of coding frames in a GOP using the optimal set of quantization parameters. An example of a trellis graph for temporal scalability using hierarchical B-Picture for the first GOP sized eight is shown in Fig. 3.

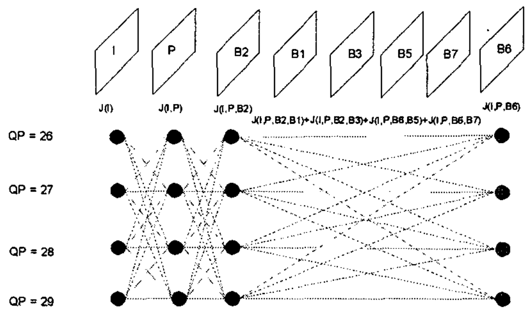

Fig. 3. Trellis graph for hierarchical B-Pictures in the first GOP sized 8.

The complexity of searching the optimal path grows exponentially when the number of nodes and branches increase in the trellis graph. To cope with this problem, we use the monotonicity property to reduce the number of nodes and branches requires for searching [2]. The full searching algorithm is shown below.

Step 1: Generate J(I) for all quantizers $q_I \in Q$.
Step 2: (Monotonicity) Prune out all I-nodes lying below minimum cost node $q_I^* \in Q$ in Step 1.
Step 3: Grow J(I,P) for all combinations of $q_P \in Q$ and all remaining $q_I \in Q$ after Step 2.
Step 4: (Monotonicity) Prune out suboptimal I-P combinations of $q_P \in Q$ and $q_I \in Q$.
Step 5: Repeat Step 3 and Step 4 for B2 and B6.
Step 6: For every surviving I-P-B4-B2-B6 combination, find the B1, B3, B5,and B7 best quantizer set that minimizes J(B1)+J(B3)+J(B5)+J(B7), i.e., populate the branch costs of trellis, because the Q values of these frames do not affect any other frames.
Step 7: Best Q combination for I-P-B4-B2-B6-B1-B3-B5-B7 is achieved.

The optimal path in Algorithm 1 is the solution for the quantization parameter for frames in the GOP. After getting the best quantization parameter in the first GOP, we freeze the solution for the current GOP and we go to the next GOP. Hence, we cut the dependency into one GOP. The reason of this is to decrease the depth of the dependency so that the computational complexity is reduced. However, we will get the sub-optimal solution as a result.

3.3. Lagrange Multiplier Selection

The next problem is how to determine the best Lagrange multiplier λ. There are several methods that can be used for determining Lagrange multiplier, one of them is bisection search [3]. Nonetheless, it is very time consuming to try to find the exact Lagrange multiplier, and the rate of convergence is very dependent toward the initial value of Lagrange multiplier. Therefore, some papers suggest to approximating the most efficient value for Lagrange multiplier based on some experimental results [5], [7]. It was found that there is a relationship between Lagrange multiplier λ and the quantization parameter that used to quantize the video sequence. The value for efficient Lagrange multiplier λ for MPEG AVC/H.264 is shown in Eq. (7) [7]. The λ in Eq. (7) is used in [7] for determining the best mode e.g. INTRA, INTER16x16, etc, that should be applied for each macroblock within a frame by calculating the similar cost function in Eq. (6). By doing this, each macroblock will have the best R-D so that it will lead to bit-rate efficiency in term of storage and transmission.

$$\lambda_{mode} = 0.85 \times 2^{(QUANT-12)/3} \qquad (7)$$

For SVC case, the JSVM reference software uses the same Lagrange multiplier as in Eq. (7). Therefore, we will apply Eq. (7) to our proposed dependent quantization.

4. Experimental Results

In this section, we will present our implementation results for dependent quantization that is integrated in JSVM reference software.

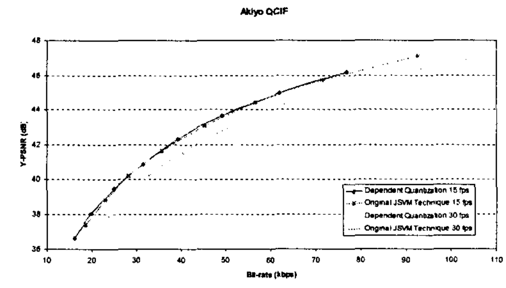

Fig. 4. PSNR to Bit-rate comparison for Akiyo QCIF sequence.

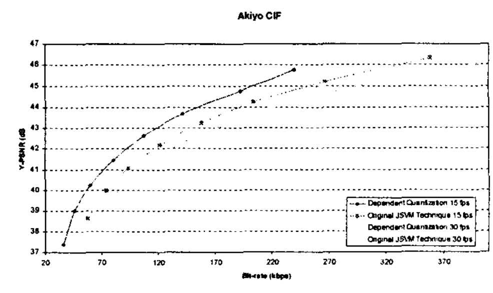

Fig. 5. PSNR to Bit-rate comparison for Akiyo CIF sequence.

Fig. 4 and 5 show the comparison of PSNR to bit-rate graph produced by current JSVM technique and the dependent quantization for Akiyo QCIF (base layer) and CIF (first EH layer of spatial scalability) sequence respectively. From Fig. 4 and 5, we can see that in the enhancement layer, the dependent quantization gains significant bit-rate reduction, while in the base layer, the bit-rate produces by the dependent quantization is offset with the current JSVM software reference technique. Fig. 6 and 7 for the News sequence shows similar results. This is because the accumulation of bit-rate reduction only big enough for the enhancement layer where we have up to 20% bit saving and up to 1.25 dB increment in PSNR values. In term of computational complexity, the dependent quantization has two/three times computational complexity compared to the current JSVM reference software technique.

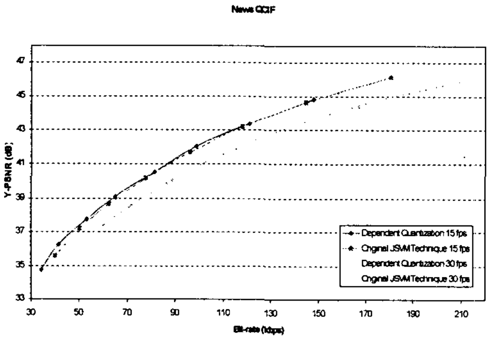

Fig. 6. PSNR to Bit-rate comparison for News QCIF sequence.

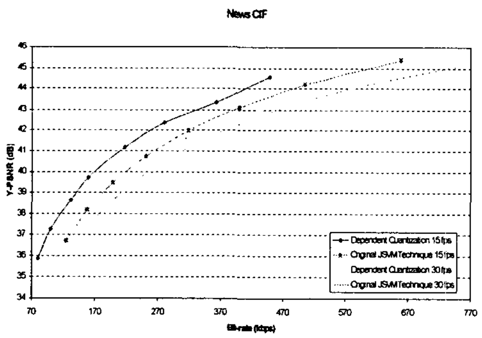

Fig. 7. PSNR to Bit-rate comparison for Akiyo CIF sequence.

5. Conclusion

In this paper, we present dependent quantization for SVC. The dependency quantization allows us to find the best combination of QP set for frames within GOP. We use the Lagrange multiplier defined in [7] and the optimization technique is done by using trellis graph to determine the best QP set. In order to reduce the complexity, we also incorporate pruning using monotonicity property.

We implemented the dependent quantization with JSVM reference software. The experimental result shows that the dependent quantization gains bit-rate reduction in the enhancement layer up to 20% bit saving and 1.25 dB of PSNR increment compared to current JSVM reference software technique with slightly increment in computational complexity. This provides more possibility for clients with constrained bandwidth to consume the bit-stream of the enhancement layer of the SVC. Furthermore, we can achieve more efficient bit-stream for storage and transmission.

References

1. ITU-T and ISO/IEC JTC1, "Joint Scalable Video Model JSVM-5," JVT-Q024, 2006.
2. K. Ramchandran, A. Ortega, and M. Vetterli, "Bit Allocation for Dependent Quantization with Applications to MPEG Video Coders," IEEE Transaction on Image Processing, Vol. 3, No. 5, Sept. 1994.
3. Liang-Jin Lin, "Video Bit-Rate Control with Spline Approximated Rate-Distortion Characteristics,"Doctoral Dissertation for Faculty of the Graduate School of University of Southern California, 1997.
4. T. Wiegand, M. Lightstone, D. Mukherjee, T.G. Campbell, and S.K. Mitra, "Rate-Distortion Optimized Mode Selection for Very Low Bit Rate Video Coding and the Emerging H.263 Standard,"IEEE Transactions on Circuits and Systems for Video Technology, Vol. 6, No.2, Apr. 1996.
5. G.J. Sullivan and T. Wiegand, "Rate-Distortion Optimization for Video Compression," IEEE Signal Processing Magazine, pp. 74-90, Nov. 1998.
6. T. Wiegand and B. Girod, "Lagrange Multiplier Selection in Hybrid Video Coder Control," IEEE International Conference on Image Processing, 2001.
7. T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G.J. Sullivan, "Rate-Constrained Coder Control and Comparison of Video Coding Standards," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, No. 7, Jul. 2003.