Title:	**Lagrange Multiplier Selection for Rate-Distortion Optimization in SVC**
Status:	Input Document to JVT
Purpose:	Proposal

Author(s) or Contact(s):

Xiang Li, Peter Amon, Andreas Hutter
Siemens Corporate Technology
Otto-Hahn-Ring 6,
81737, Munich
Germany

Tel:
Email: li.xiang.ext@siemens.com
p.amon@siemens.com
andreas.hutter@siemens.com

André Kaup
University of Erlangen-Nuremberg
Cauerstraße 7
D-91058 Erlangen
Germany

kaup@lnt.de

Source: Siemens Corporate Technology

Abstract

The Lagrange multiplier based rate-distortion optimization (RDO) has been widely employed in single layer video coding. During the development of scalable video coding (SVC) extension of H.264/AVC, it was directly applied in a multi-layer scenario. However, such an application is not very efficient since the correlation between layers is not considered in the Lagrange multiplier selection. To improve the overall performance, in this contribution a new selection algorithm is presented for RDO in SVC. Simulations show that the proposed method outperforms the most recent SVC reference software (JSVM 9.15). With small computational cost, average gains of 0.22 dB and 0.35 dB were reportedly achieved in the tests of four-layer quality scalability and three-layer spatial scalability, respectively.

1. Introduction

Rate-Distortion Optimization (RDO) techniques have been widely used in today's single layer video coding. Its target is to minimize the distortion D for a given rate R_c by appropriate selections of coding parameters, namely

$$\begin{gathered}\min\{D\}\\ \text{subject to } R \leq R_c\end{gathered} \tag{1}$$

To solve such a kind of constrained problem, Lagrange multiplier method is normally used due to its lower computational complexity. Then (1) is converted to

$$\begin{gathered}\min\{J\}\\ \text{where } J = D + \lambda \cdot R\end{gathered} \tag{2}$$

where J denotes the Lagrangian cost function and λ is called Lagrange multiplier. Consequently, how to determine λ becomes a key problem in Lagrangian RDO.

Supposing R and D to be differentiable everywhere, the minimum cost J is given by setting its derivative to zero, i.e.,

$$\frac{\mathrm{d}J}{\mathrm{d}R} = \frac{\mathrm{d}D}{\mathrm{d}R} + \lambda = 0 \tag{3}$$

leading to

$$\lambda = -\frac{\mathrm{d}D}{\mathrm{d}R} \tag{4}$$

Essentially, (4) forms the basis of the Lagrange multiplier selection method since it indicates that λ corresponds to the negative slope of the rate-distortion (R-D) curve.

Assuming a sufficiently high rate environment, [1] proposed the rate model R_S and distortion model D_S for single layer video coding, namely

$$\begin{aligned} R_S &= a \log_2(\frac{b}{D_S}) \\ D_S &= \frac{Q^2}{12} \end{aligned} \tag{5}$$

where a and b are two constants, Q is the quantization step.

When putting (5) into (4), λ for a single layer can be determined by

$$\lambda = -\frac{\mathrm{d}D}{\mathrm{d}R} = c \cdot Q^2 \tag{6}$$

where c is a constant which is experimentally suggested to be 0.85 [1], though others proposed 0.68 [2].

In fact, (6) is the Lagrange multiplier selection method in the current H.264/AVC reference software JM14.1 [3]. Moreover, during the development of scalable video coding (SVC) extension of H.264/AVC, it was also applied in the joint scalable video model (JSVM) [4]. However, such a direct application of (6) in a SVC scenario is not very efficient since the correlation between layers is not considered in (6). Therefore in this contribution, a new λ selection algorithm is proposed in this contribution.

2. Lagrange Multiplier Selection in SVC

In this section, the proposed λ selection method is first discussed. Then the implementation details are provided.

2.1. Multi-Layer Lagrange Multiplier Selection

In a SVC scenario, all the layers should be jointly optimized in order to achieve the best performance. Without loss of generality, a two-layer scenario is studied in this contribution for simplicity. In such a case, the task of RDO is extended as [5].

$$\min\{J\} = \min\{(1-w) \cdot J_0 + w \cdot J_1\} \tag{7}$$

where J_0 and J_1 represent R-D cost for the base layer and enhancement layer, and $w \in [0,1]$ is a layer weighting factor. Intuitively, when $w=0$ or $w=1$, (7) reduces to the single layer RDO for the base layer or the enhancement layer, respectively.

More concretely, J in (7) is written to

$$J = (1-w) \cdot (D_0 + \lambda_0 \cdot R_0) + w \cdot (D_1 + \lambda_1 \cdot (R_0 + R_1)) \tag{8}$$

where R_0 is included in the second term (the calculation of J_1) since in SVC the accumulated bitrate instead of self bitrate for a layer is used to evaluate the coding efficiency.

Similar to the single layer RDO, the minimum J in (7) can be obtained by setting its derivative to zero. To calculate this, R-D models are needed. Besides those in (5), there are other models proposed in the literature, such as the Laplace distribution based models in our previous work [6]. Although our Laplacian models show a better accuracy, those in (5) are employed in this contribution to ease the comparison with the reference software and demonstrate the gain is directly from the proposed algorithm.

Considering (5) actually describes the rate and distortion at pixel level and no frame size information is counted, a resolution factor has to be introduced when applying these models in a spatial scalability scenario. Therefore, we propose to calculate the joint cost J as

$$J = (1-w)\cdot(D(Q+\Delta)+\lambda_0\cdot R(Q+\Delta)) + w\cdot(D(Q)\cdot\beta+\lambda_1\cdot(R(Q+\Delta)+R(Q)\cdot\beta)) \quad (9)$$

where $D(\cdot)$ and $R(\cdot)$ represent the models in (5), $(Q+\Delta)$ and Q are quantization steps for the base layer and enhancement layer, respectively, and β denotes the resolution ratio between the two layers. For example, $\beta=1$ describes a same-resolution case, while $\beta=4$ may indicate a QCIF-CIF environment. As well known, for the same content at different resolutions, the bitrate ratio may be approximated by the resolution ratio although the two ratios are not precisely the same. In this contribution, such an approximation is directly employed for simplicity. For further improvement, more accurate models can be applied to compensate this problem.

Putting (5) into (9) and then setting the derivative of J to zero, λ_1 can be solved as

$$\lambda_1 = \frac{Q((1-w)(\ln(2)(Q+\Delta)^2 - 12a\lambda_0) + \ln(2)\beta w Q(Q+\Delta))}{12aw(\beta(Q+\Delta)+Q)} \quad (10)$$

If λ_0 is determined by the single layer λ selection method (6), i.e., plugging $\lambda_0 = c\cdot(Q+\Delta)^2$ into (10), λ_1 is derived as

$$\lambda_1 = \frac{\beta\cdot(Q+\Delta)}{\beta\cdot(Q+\Delta)+Q}\cdot(c\cdot Q^2) \quad (11)$$

Considering the last term in (11) is actually the λ by the single layer selection method (6), λ_1 is written to

$$\lambda_1 = \Gamma\cdot\lambda(Q) \quad (12)$$

where Γ is defined in (13) and $\lambda(Q)$ denotes the single layer λ function by (6).

$$\Gamma = \frac{\beta\cdot(Q+\Delta)}{\beta\cdot(Q+\Delta)+Q} \quad (13)$$

According to the definition in H.264/AVC [7], the quantization steps for the base and enhancement layers are derived as

$$\begin{aligned} Q_0 &= Q+\Delta = 2^{(QP_0-12)/6} \\ Q_1 &= Q = 2^{(QP_1-12)/6} \end{aligned} \quad (14)$$

where QP_0 and QP_1 are quantization parameters for the base and enhancement layers, respectively.

Taking (14) into (13), Γ is simplified to

$$\Gamma = \frac{\beta\cdot 2^{\Delta QP/6}}{\beta\cdot 2^{\Delta QP/6}+1} \quad (15)$$

where $\Delta QP = QP_0 - QP_1$ which describes the difference between the quantization parameters for the two layers.

Fig. 1 shows two ΔQP-Γ curves with different β. For a given β, Γ is monotonously increasing with ΔQP. When ΔQP is very big, Γ approaches 1 and λ_1 approaches λ of single layer. This is reasonable since a very big ΔQP will result in a very low correlation between the base and enhancement layers so that the optimal solution is to optimize the two layers independently. Contrariwise, when ΔQP is too small, Γ approaches 0, which will lead to a very small λ_1. In practice, it is a very rare condition since the PSNR quality of the enhancement layer will be much worse than the base layer due to the much larger quantization step. In such a case, the rate of the enhancement layer is negligible when compared with that of the base layer, and the quality is much more important. Thus a very small λ_1 is necessary. On the other hand, in the environment of spatial scalability, bigger β will lead to a bigger Γ. This is also easy to understand. When the resolution of the enhancement layer is much higher than that of the base layer, more weighting is put on the enhancement layer. Consequently, the joint optimization is shifted towards the independent RDO for the enhancement layer.

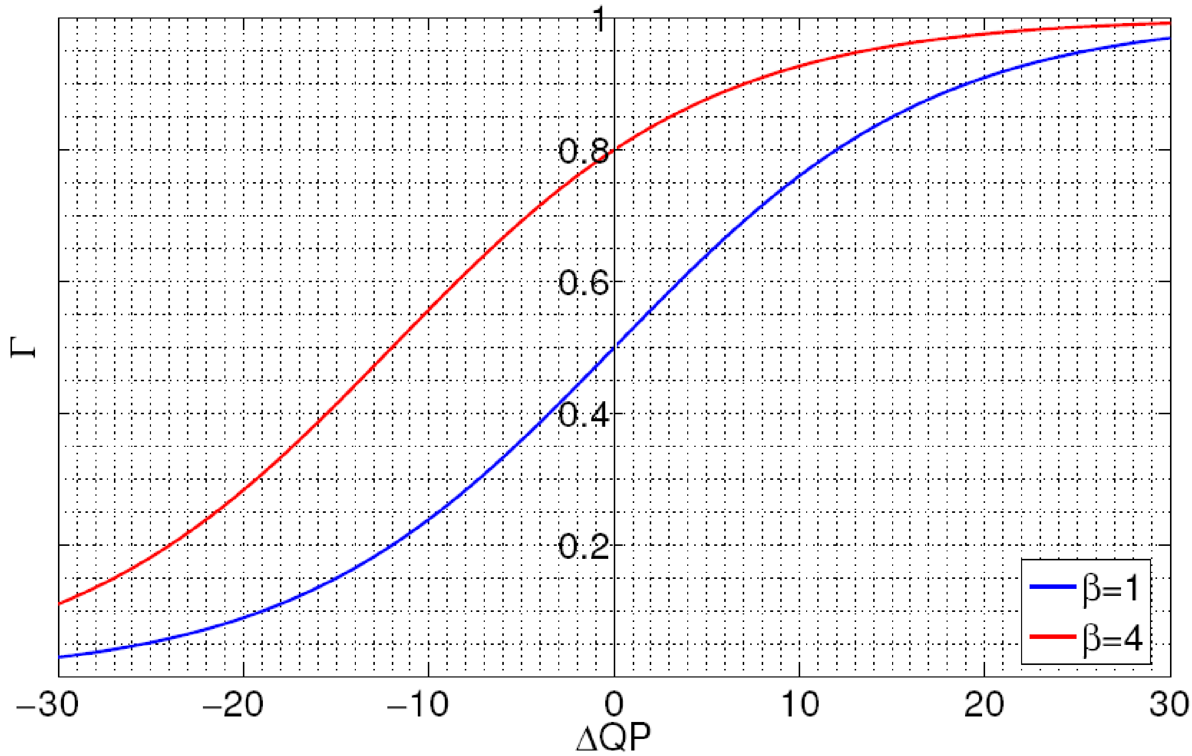

Fig. 1 Relationship among Γ, ΔQP and β

2.2. Implementation Details

Basically, (12) formulates the proposed multi-layer Lagrange multiplier selection method ML-λ where a new factor Γ is introduced. (15) shows that the computational complexity for Γ is quite low. Considering all the calculations in (12) occur at frame level, the total computational cost by the proposed algorithm is rather marginal.

Although the proposed algorithm is derived in a two-layer scenario, it can be easily extended to a multi-layer environment by counting the costs of other layers in (7). However in practice, there is an even simpler way. Since simulations indicate that the correlation between two non-neighboring layers is comparatively low, λ_n normally shows little impact on the layer $n+2$ and higher layers. Therefore a two-layer sliding window process is employed. That is for layer n, Γ_n is first derived based on ΔQP and β between its base layer and itself. Then λ_n is calculated according to (12). Since there is no base layer for layer 0, in the current implementation λ_0 is unchanged. Consequently, the performance of the layer 0 is exactly the same as that of the JSVM software.

As mentioned in Introduction, there were two values proposed for the constant c in (6). To cover the both cases, Γ' is defined as in (16), i.e., Γ and Γ' correspond to $c = 0.85$ and $c = 0.68$, respectively.

$$\Gamma' = \frac{0.68}{0.85} \cdot \Gamma = 0.8 \cdot \Gamma \tag{16}$$

In the following simulations, the performance of ML-λ with both Γ and Γ' will be evaluated. For convenience, they will be referenced as ML-$\lambda(\Gamma)$ and ML-$\lambda(\Gamma')$, respectively.

3. Simulation Results

The proposed algorithm was verified by the most recent SVC reference software JSVM 9.15 [8] in the environments of quality scalability and spatial scalability. Basically, CABAC, fast search algorithm for motion estimation were enabled, while middle granularity scalability, 8x8 transform, and low complexity MB mode were disabled. To evaluate the overall coding efficiency, the gain in PSNR-Y over the JSVM software is calculated according to [8].

3.1. Quality Scalability (IPPP)

In this sub-section, the performance of the proposed algorithm is evaluated in the environment of quality scalability (IPPP). Eight sequences defined in [9] are coded. To cover a practical quality range, four layers are employed with fixed quantization parameters, i.e., $QP = (36, 32, 28, 24)$. Clearly in this test $\Delta QP = 4$ and $\beta = 1$. According to (15) and (16), $\Gamma = 0.61$ while$\Gamma' = 0.49$.

Attached file QualityScalability_IPPP.xls presents the simulation results for quality scalability. On average, 0.19 dB and 0.22 dB gains over JSVM software are achieved by ML-$\lambda(\Gamma)$ and ML-$\lambda(\Gamma')$, respectively.

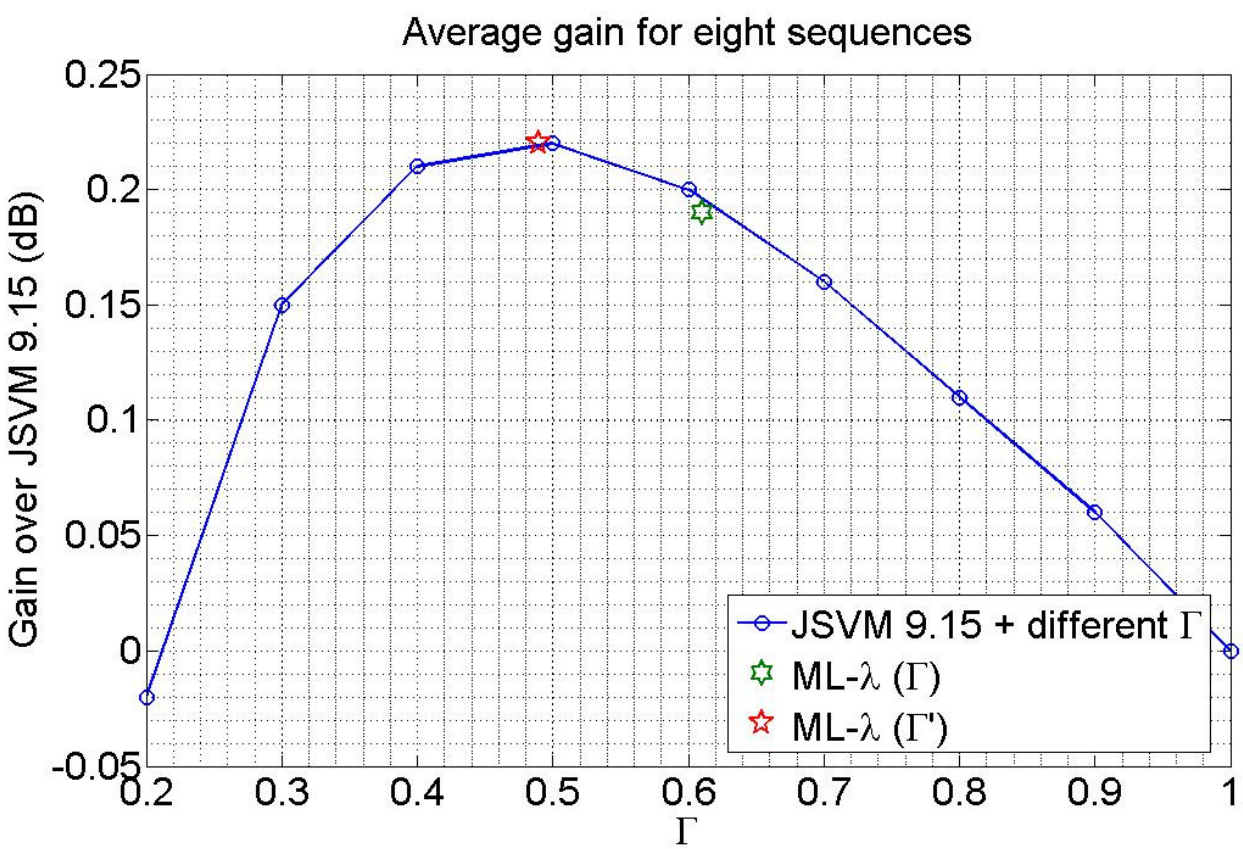

Fig. 2 Performance of different Γ values in quality scalability

Moreover, to check whether the theoretical value $\Gamma = 0.61$ ($\Gamma' = 0.49$) is optimal, simulations on different Γ values were conducted. Fig. 2 investigates the relationship between Γ value and the average gain for the same eight sequences. As shown in the figure, when $\Gamma' = 0.49$, the best performance is achieved, which well matches the theory. In addition, the performance of ML-$\lambda(\Gamma)$ is a little worse than ML-$\lambda(\Gamma')$, which indicates in quality scalable environment 0.68 is a better value than 0.85 for c in (6).

3.2. Spatial Scalability (IPPP)

In this sub-section, the performance of the proposed algorithm is evaluated in the environment of spatial scalability (IPPP). Four sequences defined in [9] are tested in a similar way to [5], i.e., each of them is coded into three spatial layers where $QP_0 = (34, 30, 26, 22)$, $QP_1 = (36, 32, 28, 24)$, and $QP_2 = (38, 34, 30, 26)$. In such an environment, $\Delta QP = -2$ and $\beta = 4$. Consequently, $\Gamma = 0.76$ while $\Gamma' = 0.61$.

Attached file SpatialScalability_IPPP.xls presents the simulation results for spatial scalability where Ln-PSNR and Sum-PSNR represent the gain for the layer n and the sum gain for the three layers, respectively. Since the layer 0 is identically coded with JSVM, L0-PSNR is always zero. On average, 0.25 dB and 0.35 dB gains over JSVM 9.15 are obtained by ML-$\lambda(\Gamma)$ and ML-$\lambda(\Gamma')$, respectively.

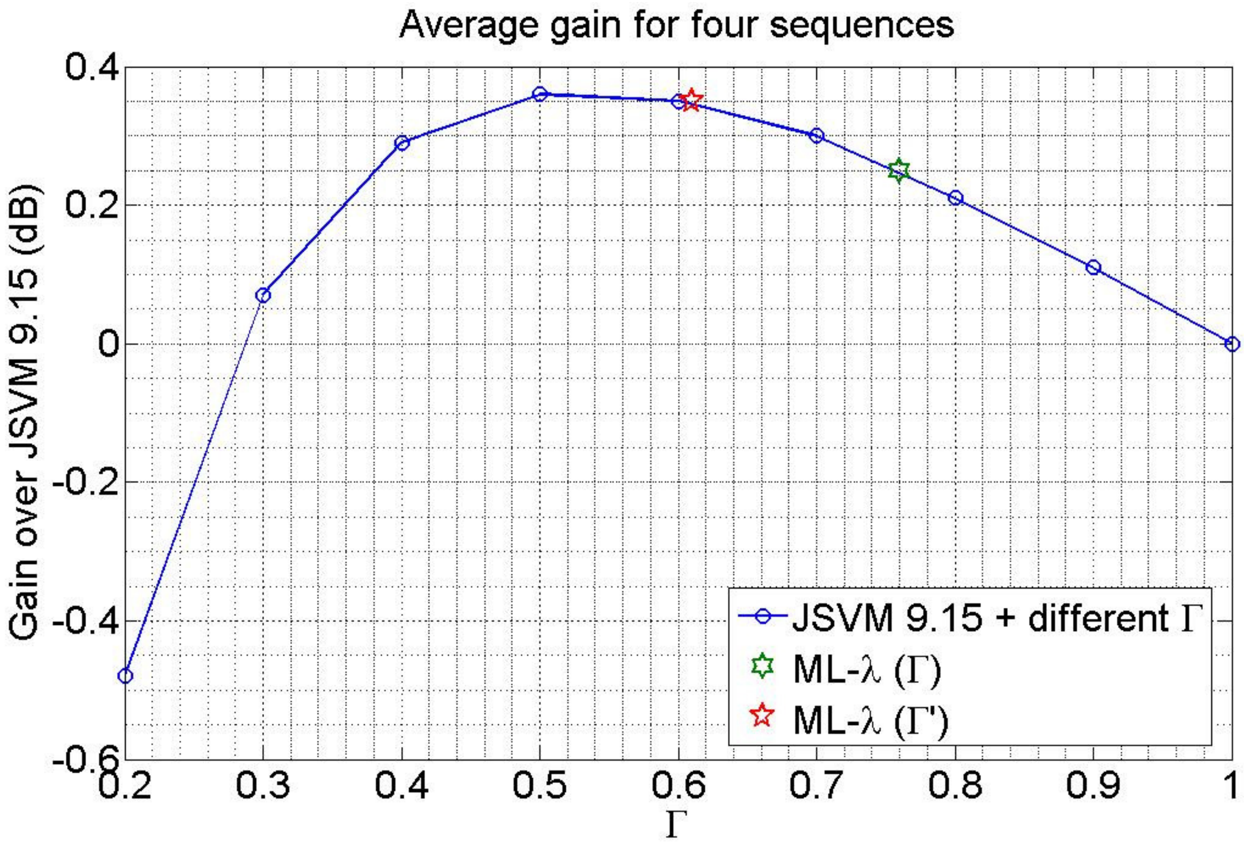

Fig. 3 Performance of different Γ values in Spatial Scalability

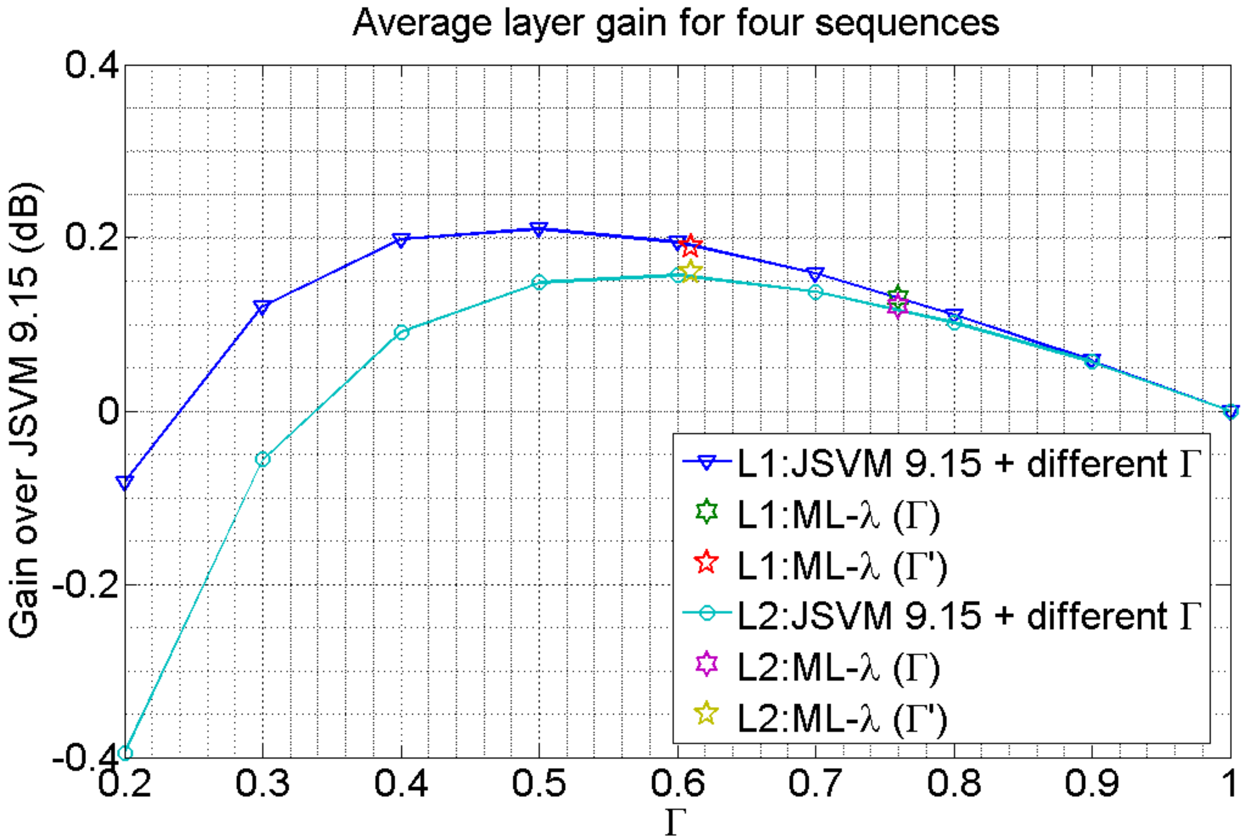

Fig. 4 Average gain for layers in spatial scalability

Fig. 3 and Fig. 4 show relationship between Γ value and the average Sum-PSNR. Again, the best performance is obtained around $\Gamma' = 0.61$, which indicates the proposed theory works well in spatial scalable scenario.

3.3. Quality Scalability (H-B)

In this sub-section, the performance of the proposed algorithm is evaluated in the environment of quality scalability with hierarchical-B (GOP=16). Eight sequences defined in [9] are tested in a similar way to [5], i.e., each of them is coded into two quality layers where $QP_0 = (40, 36, 32, 28)$, and $QP_1 = (34, 30, 26, 22)$. In such an environment, $\Delta QP = 6$ and $\beta = 1$. Consequently, $\Gamma = 0.67$ while $\Gamma' = 0.53$.

Attached file QualityScalability_HB.xls presents the simulation results for quality scalability. On average, 0.012 dB and -0.028 dB gains over JSVM software are achieved by ML-$\lambda(\Gamma)$ and ML-$\lambda(\Gamma')$, respectively. Compared with IPPP coding structure, the proposed algorithm gains little for H-B scheme. It is mainly because in H-B coding, inter-layer prediction is less frequently used since temporal prediction from the same layer is quite efficient for B frames. In such a case, the proposed algorithm which focuses the correlation between layers can only provide a similar performance to the reference software.

3.4. Spatial Scalability (H-B)

In this sub-section, the performance of the proposed algorithm is evaluated in the environment of spatial scalability with hierarchical-B (GOP=16). Four sequences defined in [9] are tested in a similar way to [5], i.e., each of them is coded into two spatial layers where $QP_0 = (34, 30, 26, 22)$, and $QP_1 = (38, 34, 30, 26)$. In such an environment, $\Delta QP = -4$ and $\beta = 4$. Consequently, $\Gamma = 0.72$ while $\Gamma' = 0.57$.

Attached file SpatialScalability_HB.xls presents the simulation results for spatial scalability. On average, 0.05 dB gain over JSVM 9.15 are obtained by ML-$\lambda(\Gamma)$ and ML-$\lambda(\Gamma')$. The reason for the marginal gain is the same as that in the last subsection, i.e., the interlayer prediction is less frequently used in H-B coding structure.

4. Conclusions

In this contribution, a Lagrange multiplier selection method ML-λ is proposed for the rate-distortion optimization in SVC. By joint consideration of base and enhancement layers, a new factor Γ was introduced as a supplementary multiplier to the single layer λ. Compared with the current algorithm in most recent SVC reference software, the new algorithm achieves a better coding efficiency at a negligible computational cost, especially for IPPP coding structure. Therefore, we propose to include this algorithm into the reference software JSVM.

5. References

1. T. Wiegand and B. Girod, “Lagrange multiplier selection in hybrid video coder control,” in *IEEE Int. Conf. on Image Process. (ICIP)*, Thessaloniki, Greece, 2001, pp. 542–545, vol.3.
2. K. Takagi, “Lagrange multiplier and RD-characteristics (JVT-C084)”, in *JVT Meeting (Joint Video Team of ISO/IEC MPEG & ITU-T VCEG)*, 2002.
3. JVT, “H.264/AVC reference software (JM14.1),” http://iphome.hhi.de/suehring/tml/, Jul. 2008.
4. JVT, “H.264/SVC reference software (JSVM 9.15) and manual,” CVS sever at garcon.ient.rwth-aachen.de, Sep. 2008.
5. H. Schwarz and T.Wiegand, “Further results for an rd-optimized multiloop svc encoder (JVT-W071),” in *JVT Meeting (Joint Video Team of ISO/IEC MPEG & ITU-T VCEG)*, 2007.
6. X. Li, N. Oertel, A. Hutter, and A. Kaup, “Laplace distribution based Lagrangian rate distortion optimization for hybrid video coding,” *IEEE Trans. Circuits Syst. Video Technol.*, accepted in Jun. 2008.
7. JVT, Advanced Video Coding (AVC) - 3rd Edition, ITU-T Rec. H.264 and ISO/IEC 14496-10 (MPEG-4 Part 10). 2004.
8. G. Bjontegaard, “Calculation of average PSNR differences between RD-curves (VCEG-M33),” *in VCEG Meeting (ITU-T SG16 Q.6)*, Austin, Texas, USA, Apr. 2001.
9. M. Wien and H. Schwarz, “Testing Conditions for SVC Coding Efficiency and JSVM Performance Evaluation (JVT-Q205),” in *JVT Meeting (Joint Video Team of ISO/IEC MPEG & ITU-T VCEG)*, 2005.