Context Adaptive Lagrange Multiplier (CALM) for Rate-Distortion Optimal Motion Estimation in Video Coding

Jun Zhang, Xiaoquan Yi, *Member, IEEE,* Nam Ling, *Fellow, IEEE,* and Weijia Shang, *Member, IEEE*

***Abstract*—In this paper, we propose an efficient and practical algorithm to dynamically adapt the Lagrange multipliers for each macroblock based on the context of the neighboring or upper layer blocks to improve rate-distortion performance. Our method improves the accuracy for the detection of true motion vectors as well as the most efficient encoding modes for luma, which are used for deriving the motion vectors, and modes for chroma. Simulation results for H.264/advanced video coding video demonstrate that our method reduces bit rate significantly and achieves peak signal-to-noise ratio gain over those of the joint model (JM) software for all sequences tested, with negligible extra computational cost. The improvement is particularly significant for high motion high-resolution videos. This paper describes our work that led to our Joint Video Team adopted contribution (included in software JM 12.0 onward), collectively known as context adaptive Lagrange multiplier (CALM).**

***Index Terms*—H.264/advanced video coding (AVC), Lagrange multiplier, motion estimation, rate-distortion optimization, source coding, video coding, visual communications.**

I. Introduction

The FUNDAMENTAL issue for video coding is to achieve the highest reproduction video quality using the lowest bit rate possible. A tradeoff is made between bit rate and fidelity [1]. Typical techniques for digital compression include prediction, transformation, quantization, and entropy coding. Much of the improved compression performance can be achieved by taking advantage of the large amount of temporal redundancy in video content. ITU-T H.264 [ISO/IEC MPEG-4 advanced video coding (AVC)] [2] is a state-of-the-art international video coding standard providing powerful features. The design and operation of an encoder involves the optimization of many decisions to obtain the best possible tradeoff between rate and distortion, given the constraints on delay or complexity.

There has been a significant amount of work on the encoder optimization problem. One particular area has been the Lagrangian optimization method [3]–[7]. Many studies have developed advanced encoder optimization strategies [8]–[18]. The rate-distortion (R-D) optimization (RDO) for video encoding by using the Lagrange multiplier techniques is addressed in [9] for H.263 encoders. The Lagrange multiplier optimization technique provides a systematic way to select the optimal coding mode [10], [14] for H.264/AVC and other standards. Kannangara *et al.* [18] propose an early skip prediction for H.264/AVC by estimating a Lagrange rate-distortion cost function which incorporates an adaptive model for the Lagrange multiplier parameter based on local sequence statistics. To avoid the expensive computation of Lagrange costs, Tu *et al.* [16] and Chen *et al.* [17] suggest transform-domain bit rate estimation and distortion measures based on quantized and inverse quantized integer transform coefficients, for inter-mode decision in H.264/AVC coders.

One important decision is what motion vector (MV), or a set of block MVs, should be used for macroblock motion compensation in the inter-mode. Another important decision is the choice between inter and intra modes. For motion estimation (ME), the typical choice is based on the sum of absolute differences (SAD) or the sum of squared differences (SSD) criteria between the current frame and the reference frame which correlate with the resulting residual bit rate and distortion. However, using SAD or SSD alone does not consider bits required by encoding of MVs. Therefore, in the H.264/AVC joint model (JM) software [20], the following cost function is used in motion estimation:

$$J(\mathbf{m}, \lambda_{\mathrm{MOTION}}) = SAD(s, c(\mathbf{m})) + \lambda_{\mathrm{MOTION}} R(\mathbf{m} - \mathbf{p}) \qquad (1)$$

where $\mathbf{m} = (m_x, m_y)$ is the motion vector, $\mathbf{p} = (p_x, p_y)$ is the prediction for the motion vector, and $\lambda_{\mathrm{MOTION}}$ is the Lagrange multiplier. The rate term $R(\mathbf{m} - \mathbf{p})$ represents the bits to code predicted motion vector error. $SAD(s, c(\mathbf{m}))$ corresponds to the distortion, measured as SAD for full-pel motion estimation or sum of absolute transformed differences [SA(T)D] for sub-pel motion estimation. For an $M \times N$ block, $SAD(s, c(\mathbf{m}))$ is computed as

$$SAD(s, c(\mathbf{m})) = \sum_{x=1}^{M} \sum_{y=1}^{N} \left| s(x, y) - c(x - m_x, y - m_y) \right| \qquad (2)$$

with s being the original video block and c being the predicted video block at the position designated by $\mathbf{m}$ in the reference picture considered.

Manuscript received April 13, 2007; revised December 16, 2007, January 27, 2009, and August 10, 2009. Date of publication March 18, 2010; date of current version June 3, 2010. This paper was recommended by Associate Editor H. Yu.

The authors are with the Department of Computer Engineering, Santa Clara University, Santa Clara, CA 95053 USA (e-mail: j_zhang1999@yahoo.com; bill.qyi@gmail.com; nling@scu.edu; wshang@scu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2010.2045915

The above cost function serves two purposes: it takes the bits of motion vector into consideration, and it also converts a constrained optimization problem into an unconstrained problem. Now, the question is how to select the Lagrange multiplier to obtain the optimal rate-distortion (R-D) curve.

Typically, Lagrange multiplier controls the macroblock motion estimation and is determined experimentally by the following function [14], [20], [21]:

$$\lambda_{\mathrm{MOTION}} = 0.92 \cdot 2^{(qp-12)/6} \tag{3}$$

where qp is the quantization parameter.

The above Lagrange multiplier method provides significant rate-distortion performance improvement over the ones that use distortion cost alone. When we were developing the Simplified and Unified Multi-Hexagon (SUMH) (also known as Simplified Fast Motion Estimation or SFME) search algorithm [22], [23] with Dual-Halfway-Stop Normalized Partial Distortion Search (DHS-NPDS) [23], we found that some fast motion estimation methods have better rate-distortion performance than that of the full search for quite a few video sequences, which was further analyzed in [24]. This led us to believe that the cost function with the current Lagrange multiplier may not be accurate enough to capture the nature of motion estimation. Many researchers have studied the effect of using different Lagrange multiplier values [25] and it was observed that the Lagrange multiplier determined by quantization parameter alone is not accurate. The Lagrange multiplier needs to be adaptive to the characteristics of the blocks and be obtained through the context of the neighboring or upper layer blocks.

In this paper, we first analyze the global adjustment of Lagrange multipliers and its limitation in the next section. Section III presents our practical context adaptive Lagrange multiplier approach. In Section IV, experimental results are offered. In the last section we draw our conclusion.

II. Global Adjustment of Lagrange Multiplier

The major reason why the original Lagrange multipliers are not as effective is because they are based on the effect of quantization on the residue bits instead of the overall bits for coding a macroblock. Sometimes the motion vector difference bits constitute a significant portion of the overall bit rate, e.g., 20–30%, as plotted in Fig. 1. When the motion vector differences are too large, sometimes they do not represent the true motion.

In order to understand the impact of Lagrange multipliers on the R-D curve, we started with the following simple adjustment:

$$J(\mathbf{m}, \lambda_{\mathrm{MOTION}}) = SAD(s, c(\mathbf{m})) + F_{\mathrm{CONST}}\lambda_{\mathrm{MOTION}} R(\mathbf{m} - \mathbf{p}) \tag{4}$$

where F_{CONST} is a constant factor that adjusts the Lagrange multiplier defined by (3). In Experiment 1, we set the factor F_{CONST} to the value in $\{0.5, 0.8, 1.0, 1.5, 2.0\}$ and encoded two frames, the first frame was an I frame, the second was a P frame. The results for the *Football* CIF sequence are shown in Fig. 2. Note that we have the best R-D curve when F_{CONST} was set to 2.0.

However, for the *Stefan* CIF sequence, the peak signal-to-noise ratio (PSNR) value for lower bit rate with F_{CONST} being 2.0 is slightly worse than those with the original Lagrange value ($F_{\mathrm{CONST}} = 1.0$), using the same experiment to obtain the R-D curves, as shown in Fig. 3.

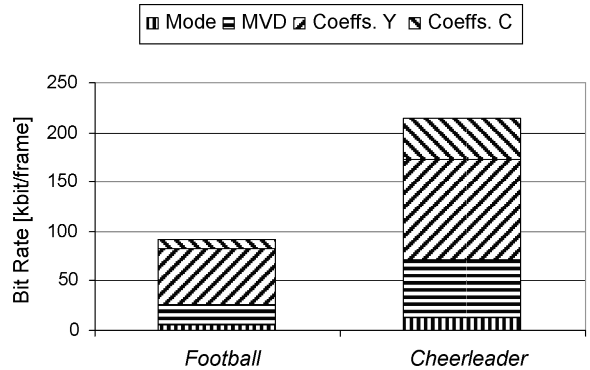


Fig. 1. Average bit rate distribution among different components for P frames. *Football* CIF 352 × 288 and *Cheerleader* D1 720 × 480, 30 frames/s, 100 frames, QP = 24, GOP structure = IPPP....

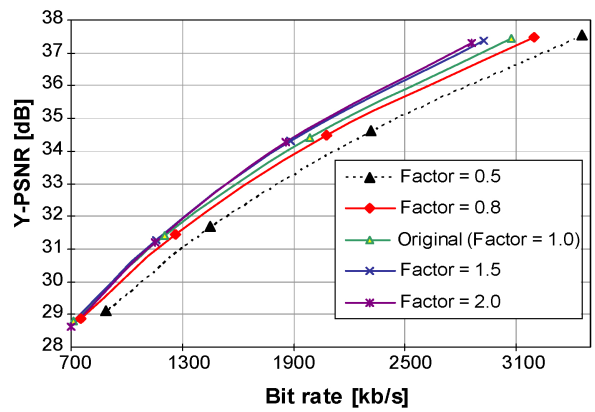


Fig. 2. *Football* CIF. QP = 24, 28, 32, 36, GOP structure = IPPP..., 30 frames/s, bits shown are for P frame.

We also explored cases when rate control is enabled. For the *Football* CIF sequence, in Experiment 2, we set the initial quantization parameter (QP) to 24 and the number of frames to 30. We also found that when $F_{\mathrm{CONST}} = 2.0$ we have the best R-D curve, as shown in Fig. 4. However, for the *Stefan* CIF sequence the PSNR value is a lot lower than that with the original Lagrange value in the lower bit rate range, as shown in Fig. 5.

Still, for the above experiments, we have only adjusted the Lagrange multiplier factor globally; that is, we used the same adjusting factor for one frame (Experiment 1) or multiple frames (Experiment 2). Although we would like to make further adjustment of the Lagrange multiplier factor at the macroblock level, it is computationally prohibitive if we were to enumerate all the combinations of the factors for each macroblock. The best way to achieve this is to approximate the optimal solution.

III. Context Adaptive Lagrange Multiplier (CALM) Method

Although the above experiments do not directly give us the optimal Lagrange multiplier factor adjustment we have the

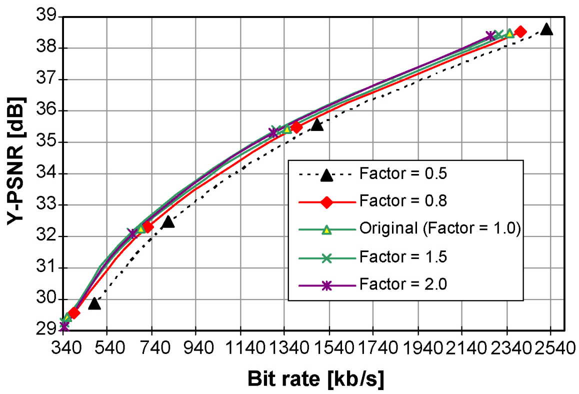


Fig. 3. *Stefan* CIF. QP = 24, 28, 32, 36, GOP structure = IPPP..., 30 frames/s, bits shown are for P frame.

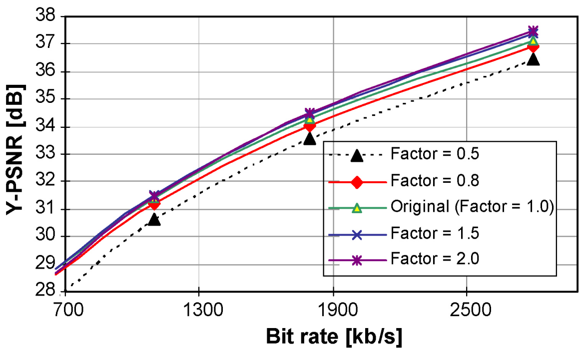


Fig. 4. *Football* CIF with rate control turned on. Initial QP = 24, GOP structure = IPPP..., 30 frames/s, bits shown are for P frame.

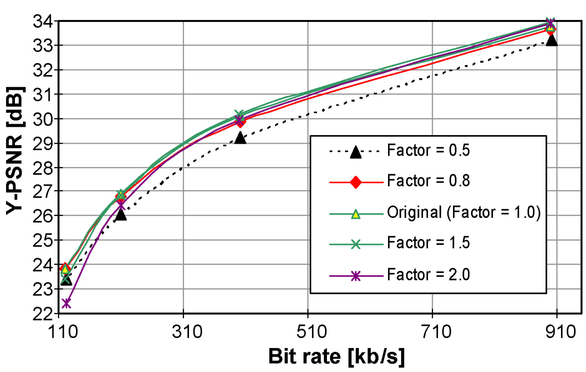


Fig. 5. *Stefan* CIF with rate control turned on. Initial QP = 24, GOP structure = IPPP..., 30 frames/s, bits shown are for P frame.

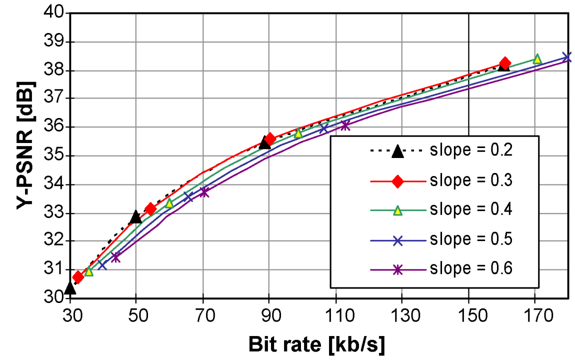


Fig. 6. *Foreman* QCIF R-D curves with different slopes for κ. QP = 24, 28, 32, 36; 30 frames/s, GOP structure = IPPP....

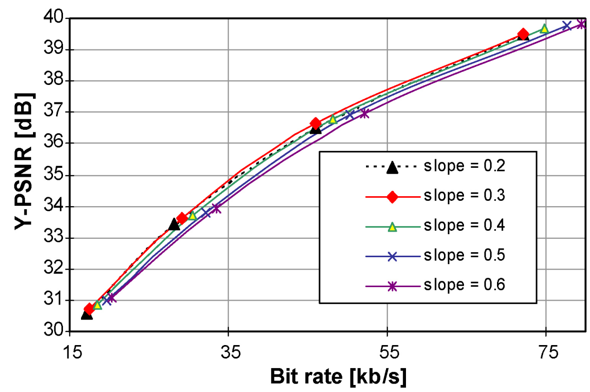


Fig. 7. *News* QCIF R-D curves with different slopes for κ. QP = 24, 28, 32, 36; 30 frames/s, GOP structure = IPPP....

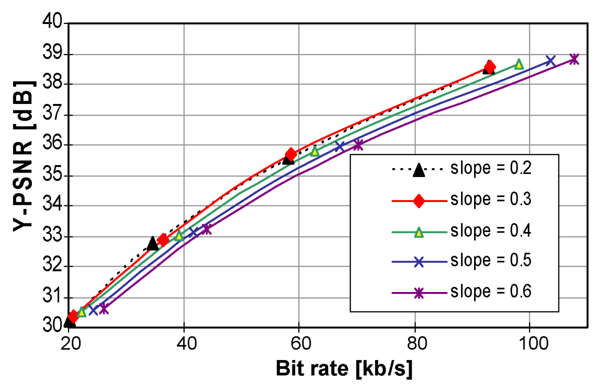


Fig. 8. *Silent* QCIF R-D curves with different slopes for κ. QP = 24, 28, 32, 36; 30 frames/s, GOP structure = IPPP....

following observations after more sequences were tested.

1) When the motions are small or constant across the entire frame or sequence, the original Lagrange multiplier already generates near-optimal R-D curves. This is consistent with our observation that the Lagrange multiplier is accurate when motion vector difference bits are low in comparison to the overall bit rate.
2) When the directions of the motion vectors in a frame are random or the motion vector differences are larger, the Lagrange multiplier needs to be increased to achieve a better R-D curve. This is exactly the case where Lagrange multiplier factor needs to be adjusted.

Based on the above observations, we propose to use the following Lagrange cost function:

$$J(\mathbf{m}, \lambda_{\text{MOTION}}) = SAD(s, c(\mathbf{m})) + F_{\text{CALM}} \lambda_{\text{MOTION}} R(\mathbf{m} - \mathbf{p}). \quad (5)$$

Our goal is to find an efficient way to determine the value of F_{CALM}, described below, for motion estimation at the macroblock level.

To determine if we need to adjust the Lagrange multiplier, we first estimate the cost $\hat{J}(\mathbf{m}, \lambda_{\text{MOTION}})$ based on (1). If it is less than a threshold J_{DEFAULT}, we would not adjust F_{CALM};

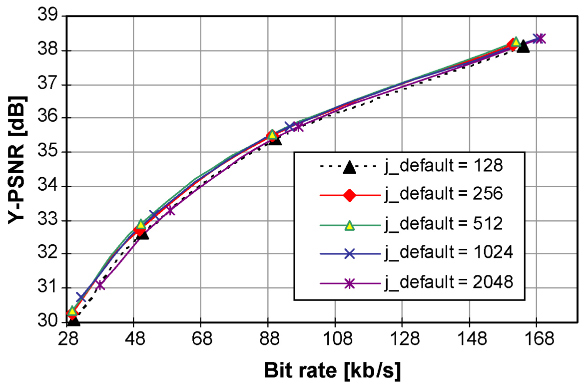


Fig. 9. *Foreman* QCIF R-D curves with different *j_default* values. QP = 24, 28, 32, 36; 30 frames/s, GOP structure = IPPP....

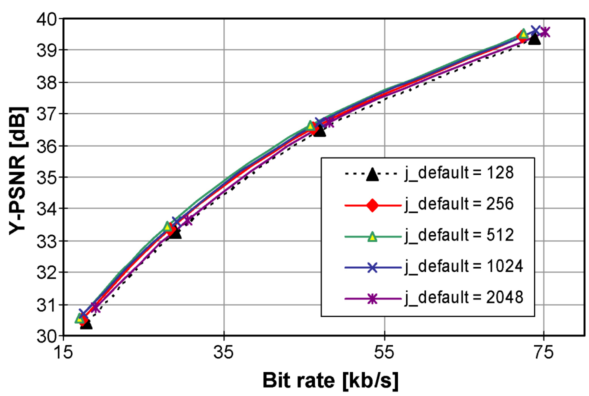


Fig. 10. *News* QCIF R-D curves with different *j_default* values. QP = 24, 28, 32, 36; 30 frames/s, GOP structure = IPPP....

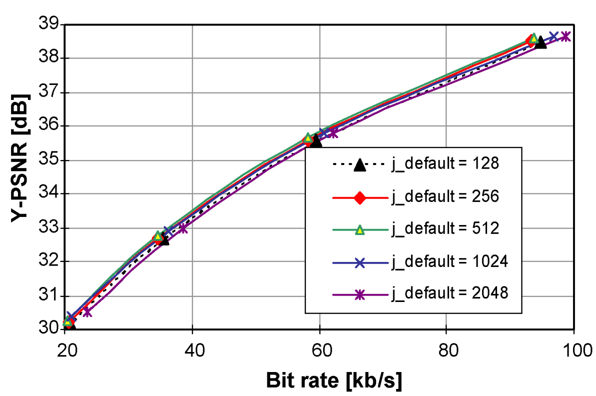


Fig. 11. *Silent* QCIF R-D curves with different *j_default* values. QP = 24, 28, 32, 36; 30 frames/s, GOP structure = IPPP....

otherwise, we would calculate F_{CALM} using the following heuristic function:

$$F_{\text{CALM}} = \sqrt{\frac{\hat{J}(\mathbf{m}, \lambda_{\text{MOTION}})}{J_{\text{DEFAULT}}\kappa(qp)}} \tag{6}$$

where $\kappa(qp)$ is a function of qp.

Since we could not obtain $\hat{J}(\mathbf{m}, \lambda_{\text{MOTION}})$ before the motion estimation of the current block, our proposed CALM method uses the Lagrange costs of neighboring macroblocks (if available) to obtain the scaling factor F_{CALM} when

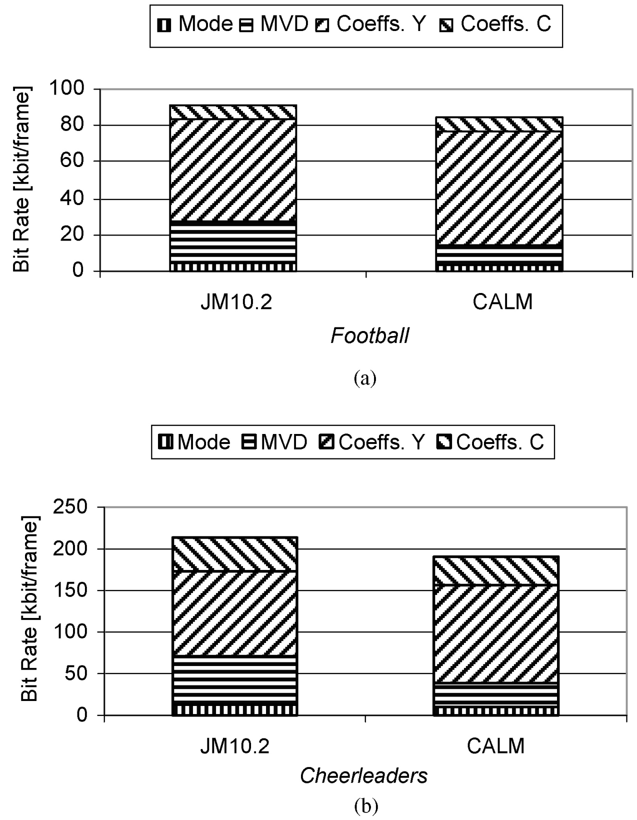


Fig. 12. Bit rate distributions among different components for P frames with QP = 24 and GOP structure = IPPP.... (a) Bit rate distributions for *Football* CIF. (b) Bit rate distributions for *Cheerleaders* D1.

performing motion estimation for the 16×16 blocks. The cost of a 16×16 block can be used for determining F_{CALM} for 16×8 and 8×16 blocks within it, whose costs can in turn be used for estimating 8×8, 8×4 or 4×8 blocks, and so on. F_{CALM} is obtained by the following pseudo-code:

For 16×16 macroblocks:
IF both left and above neighbors exist {

$$\hat{J}(\mathbf{m}, \lambda_{\text{MOTION}}) = [\hat{J}_{\text{LEFT}}(\mathbf{m}, \lambda_{\text{MOTION}}) + \hat{J}_{\text{ABOVE}}(\mathbf{m}, \lambda_{\text{MOTION}})]/2$$

} ELSEIF left neighbor exists {

$$\hat{J}(\mathbf{m}, \lambda_{\text{MOTION}}) = \hat{J}_{\text{LEFT}}(\mathbf{m}, \lambda_{\text{MOTION}})$$

} ELSEIF above neighbor exists {

$$\hat{J}(\mathbf{m}, \lambda_{\text{MOTION}}) = \hat{J}_{\text{ABOVE}}(\mathbf{m}, \lambda_{\text{MOTION}})$$

} ELSE {

$$\hat{J}(\mathbf{m}, \lambda_{\text{MOTION}}) = J_{\text{DEFAULT}}$$

}

For 16×8, 8×16, 8×8, 8×4, 4×8, and 4×4 blocks:

$$\hat{J}(\mathbf{m}, \lambda_{\text{MOTION}}) = \hat{J}_{\text{UPPER-LAYER}}(\mathbf{m}, \lambda_{\text{MOTION}})$$

IF $\hat{J}(\mathbf{m}, \lambda_{\text{MOTION}}) < J_{\text{DEFAULT}}$ {
// cost is not big enough, do not scale

$$F_{\text{CALM}} = 1.0$$

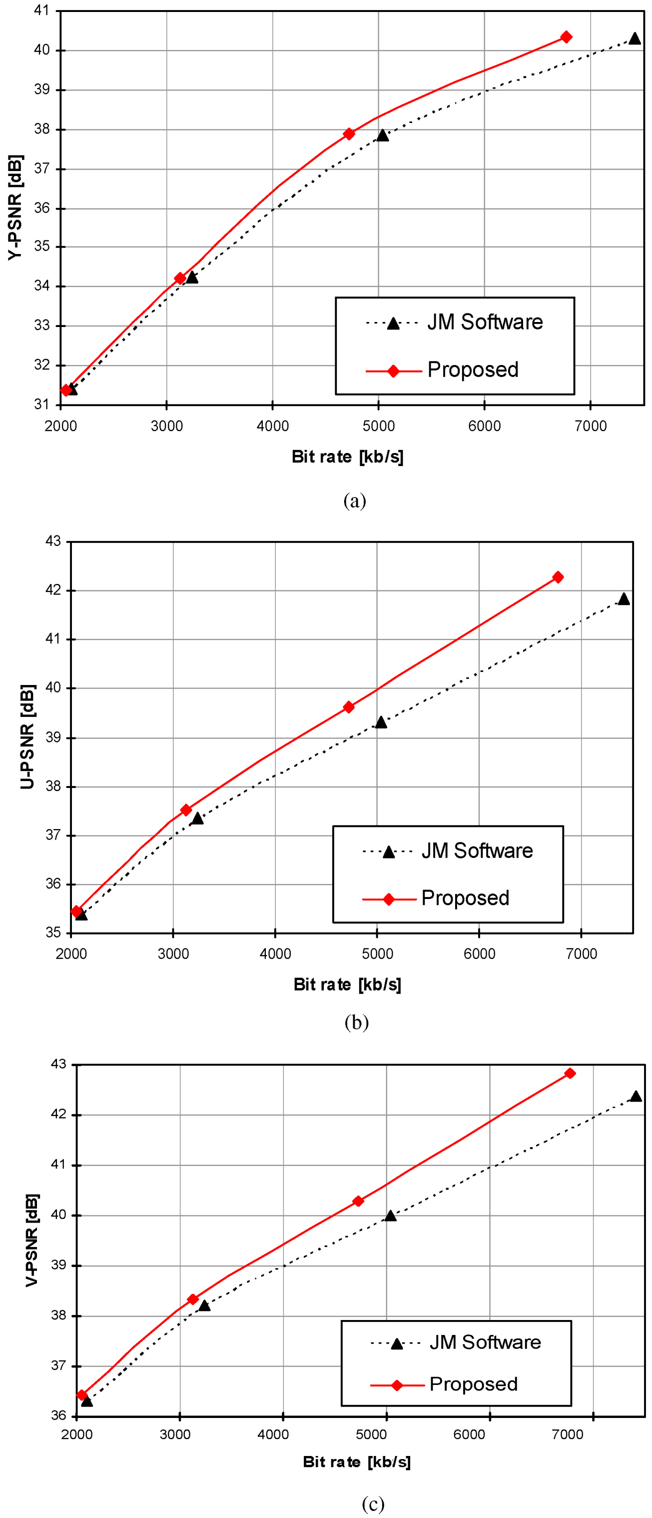


Fig. 13. Rate-distortion curve plots for Y, U, and V components for *Cheerleaders* ITU-R D1 720 × 480, 30 frames/s, 10 frames, CAVLC, GOP structure = IPPP. . . , RDO = OFF. (a) Y-component. (b) U-component. (c) V-component.

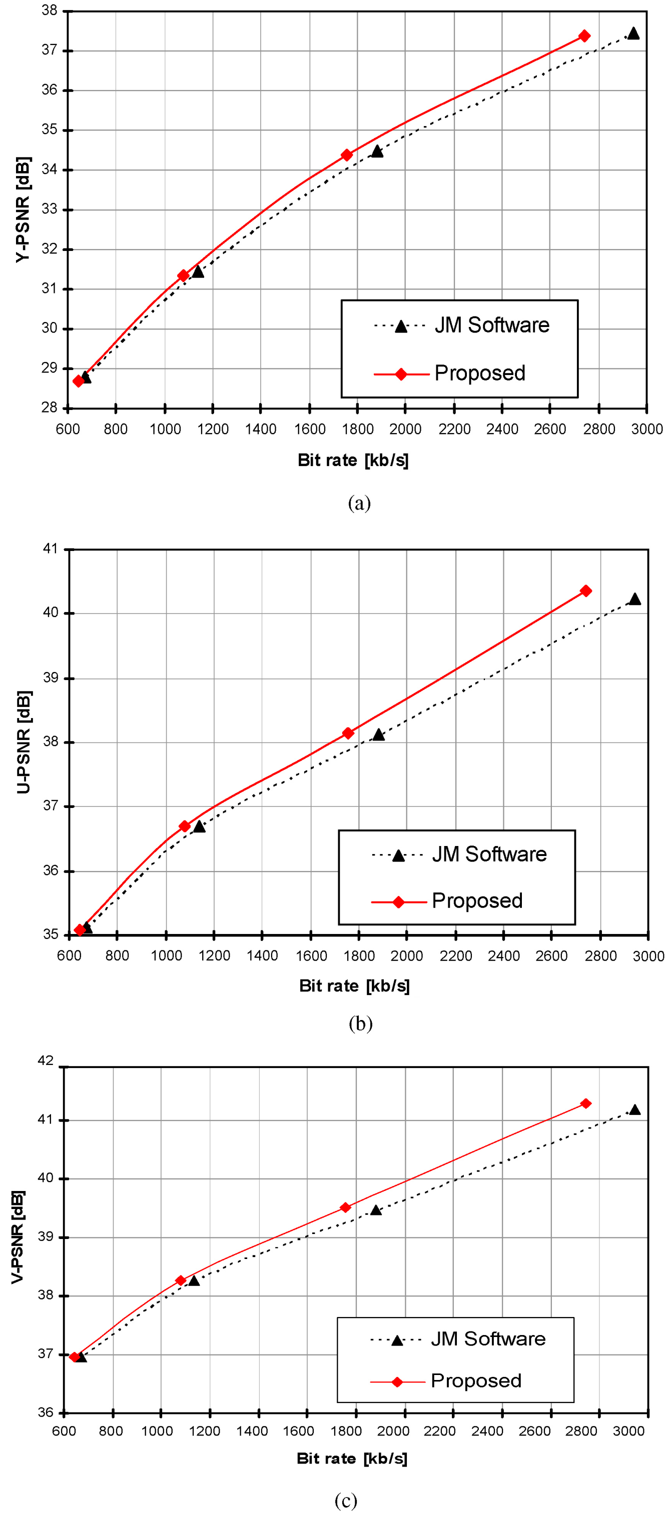


Fig. 14. Rate-distortion curve plots for Y, U, and V components for *Football* CIF 352 × 288, 30 frames/s, 100 frames, CAVLC, GOP structure = IPPP. . . , RDO = OFF. (a) Y-component. (b) U-component. (c) V-component.

} ELSE {

$$F_{\text{CALM}} = \sqrt{\frac{\hat{J}(\mathbf{m}, \lambda_{\text{MOTION}})}{J_{\text{DEFAULT}}\kappa(qp)}}$$

}

After each round of motion estimation for 16 × 16, 16 × 8, and 8 × 16 blocks, F_{CALM} is updated for the motion vector with the best estimated cost.

Based on the experiments, described below, we found that $\kappa(qp)$ should be in the range of [1.0–10.0]; it can be

TABLE I
ENCODER CONTROL PARAMETERS FOR SIMULATIONS

Name	Meaning	Value
ProfileIDC	Base/main profiles	66/77
UseHadamard	Hadamard transform	ON
SearchRange	Motion estimation search range	32–64
NumberReferenceFrames	Number of reference frames	1–3
SymbolMode	Entropy coding methods	CAVLC/CABAC
GOP structure	Group of pictures structure	IPPP
UseFME	Fast motion estimation methods	0/1/2/3
RDOptimization	Rate distortion optimized mode decision	0/1

TABLE II
EXPERIMENTAL RESULTS, REFERENCE VERSUS PROPOSED, RDO OFF (QP = 24, 28, 32, 36)

Sequences	Frame Rate	Number of Frames	Y BD bit Bit Rate Change	BD Y-PSNR Change	U BD Bit Rate Change	BD U-PSNR Change	V BD Bit Rate Change	BD V-PSNR Change
Foreman	30	100	−2.42%	0.122	−5.31%	0.134	−4.47%	0.171
News	30	100	−2.55%	0.167	−3.88%	0.176	−3.75%	0.151
Silent	30	100	−2.50%	0.139	−3.56%	0.149	−3.58%	0.131
Coastguard	30	100	−1.51%	0.066	−4.31%	0.083	0.36%	0.014
Table	30	100	−1.67%	0.087	−3.06%	0.103	−3.56%	0.145
Foreman_cif	30	100	−1.61%	0.07	−2.90%	0.071	−2.97%	0.09
Harbor_cif	30	100	−2.65%	0.125	−4.27%	0.085	−4.49%	0.078
Crew_cif	30	100	−1.26%	0.057	−1.06%	0.041	−2.08%	0.079
Stefan_cif	30	100	−3.70%	0.189	−5.36%	0.173	−5.90%	0.193
Bus_cif	30	100	−3.05%	0.165	−4.34%	0.121	−5.87%	0.166
Football_cif	30	100	−4.21%	0.263	−6.24%	0.255	−6.43%	0.212
Cheerleader_D1	30	10	−4.93%	0.361	−9.13%	0.576	−9.47%	0.567
Average			−2.67%	0.151	−4.45%	0.164	−4.35%	0.166

TABLE III
EXPERIMENTAL RESULTS, REFERENCE VERSUS PROPOSED, RDO ON (QP = 24, 28, 32, 36)

Sequences	Frame Rate	Number of Frames	Y BD Bit Bit Rate Change	BD Y-PSNR Change	U BD Bit Rate Change	BD U-PSNR Change	V BD Bit Rate Change	BD V-PSNR Change
Foreman	30	100	−0.64%	0.032	0.39%	−0.017	−0.99%	0.030
News	30	100	−0.05%	0.003	−0.43%	0.022	0.21%	−0.011
Silent	30	100	−0.28%	0.017	−0.40%	0.017	−0.34%	0.020
Coastguard	30	100	−0.29%	0.012	0.91%	−0.009	−0.02%	−0.018
Table	30	100	0.05%	−0.003	−0.61%	0.029	−0.50%	0.023
Foreman_cif	30	100	−0.45%	0.017	−0.61%	0.014	0.02%	0.002
Harbor_cif	30	100	−0.45%	0.022	−1.01%	0.018	0.02%	−0.001
Crew_cif	30	100	−0.13%	0.006	−0.41%	0.013	−0.65%	0.024
Stefan_cif	30	100	−0.10%	0.005	−0.82%	0.024	−0.31%	0.012
Bus_cif	30	100	−0.35%	0.018	−0.60%	0.017	−0.61%	0.022
Football_cif	30	100	−0.11%	0.008	−0.39%	0.013	0.08%	−0.002
Cheerleader_D1	30	10	−0.25%	0.017	−0.37%	0.025	−0.40%	0.022
Average			−0.25%	0.013	−0.36%	0.014	−0.29%	0.010

represented by the following affine function:

$$\kappa(qp) = s \cdot qp - 6.0. \tag{7}$$

To determine the slope s of (7), we tried different values for many sequences. Since it was computationally impossible to achieve global optimal R-D performance by trying different values for $\kappa(qp)$ for each step, we have decided to select some slope values and compare the R-D curves from these values. Figs. 6–8 show the R-D curves for some of the sequences, we found that when $s = 0.3$, it showed the best R-D performance for most of the sequences. When $\kappa(qp)$ is smaller than 1, we empirically set it to 1. We have also tried other functions, such as $\kappa(qp) = c_1 \times \sqrt{qp} + c_2$, where c_1 and c_2 are constants. The results were not better than those using the affine function.

For the selection of J_{DEFAULT} we tried different values around 256, which is the number of pixels in a macroblock. Figs. 9–11 show that the R-D curves with J_{DEFAULT} ranging from 128 to 2048 are not significantly different when the slope s of (7) is set to 0.3; we chose the value of 512 where the R-D curve is relatively better than those of other choices.

The values of $\kappa(qp)$ for all the qps can be stored in a lookup table, and are initialized once in the encoder for performance reason. Although square root function is used for the calculation of F_{CALM}, as the precision of this calculation

TABLE IV
EXPERIMENTAL RESULTS, REFERENCE VERSUS PROPOSED, RATE CONTROL ENABLED, RDO OFF (QP = 24, 28, 32, 36)

Sequences	Frame Rate	Number of Frames	Y BD bit Bit Rate Change	BD Y-PSNR Change	U BD bit Rate Change	BD U-PSNR Change	V BD bit Rate Change	BD V-PSNR Change
Foreman	30	100	−1.39%	0.068	−2.98%	0.078	−4.26%	0.128
News	30	100	−1.64%	0.108	−2.58%	0.104	−2.41%	0.079
Silent	30	100	−1.12%	0.065	−1.97%	0.064	−3.02%	0.093
Coastguard	30	100	−1.81%	0.079	−2.93%	0.038	−1.45%	0.027
Table	30	100	−1.00%	0.058	−0.44%	0.017	−2.94%	0.121
Foreman_cif	30	100	−1.09%	0.045	−3.79%	0.081	−4.19%	0.126
Harbor_cif	30	100	−2.66%	0.129	−4.55%	0.099	−5.47%	0.118
Crew_cif	30	100	−0.81%	0.037	−1.79%	0.054	−2.03%	0.072
*Stefan_*cif	30	100	−3.13%	0.157	−4.44%	0.131	−4.78%	0.149
Bus_cif	30	100	−2.82%	0.149	−4.63%	0.095	−5.43%	0.145
Football_cif	30	100	−3.95%	0.229	−6.05%	0.213	−6.33%	0.183
Cheerleader_D1	30	10	−4.56%	0.323	−8.15%	0.459	−8.18%	0.446
Average			−2.17%	0.121	−3.69%	0.119	−4.21%	0.141

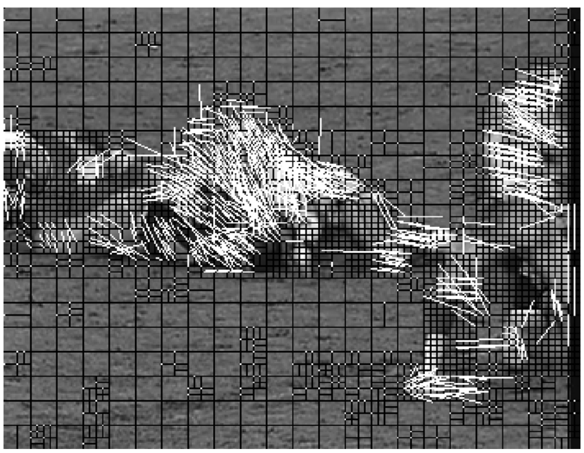


Fig. 15. Block partitions and motion vectors (in white) for *Football* CIF sequence, by JM 10.2 Reference Software, QP = 24.

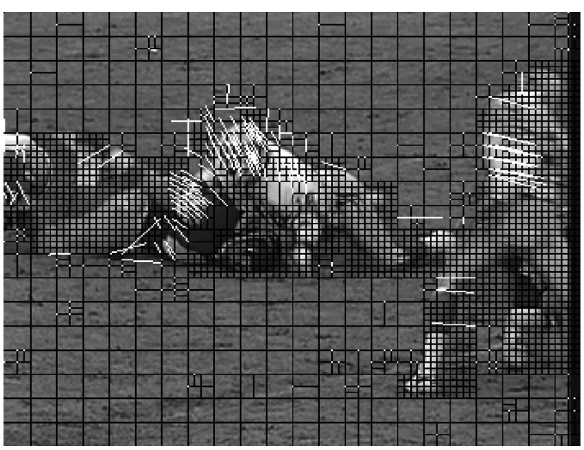


Fig. 16. Block partitions and motion vectors (in white) for *Football* CIF sequence, by CALM, QP = 24.

is not critical in this approach a fast approximation algorithm (such as Bhaskara–Brouncker algorithm, Newton's iteration, or Wolfram's iteration) can replace it if the device does not efficiently support it or lower computation is desired (such as in a parallel computing environment).

IV. EXPERIMENTAL RESULTS

A. Test Conditions and Measure Criteria

We implemented our CALM method into H.264/AVC JM 10.2 reference software [27]. A new flag CtxAdptLagrangeMult was added to enable and disable our method conveniently. We tested the proposed method in both low (RDO off) and high (RDO on) complexity modes with different motion estimation search methods for different test sequences. The Bjøntegaard Delta (BD) method [28] was used for rate-distortion performance measurement. Some common test conditions based on [29] are listed in Table I.

B. Rate-Distortion Performance Improvement

The summary of the experimental results with RDO being turned off is shown in Table II. Fast full search method [27] was used for comparison since fast full search method finds the exact same motion vectors as full search method. The rate-distortion performance improvement of our proposed algorithm is discussed in this section. The average bit rate reduction is 2.67% and the average PSNR gain is 0.151 dB, calculated using the BD method. Bit rate distributions are shown for two high motion sequences, *Football* and *Cheerleaders*, in Fig. 12. From the figure, CALM yields much less motion vector difference (MVD) bits for both *Football* (i.e., 18% versus 8%) and *Cheerleaders* (i.e., 23% versus 9%) sequences. Our proposed method performs particularly well with medium and high motion sequences for P frames. This is significant because such sequences usually require higher bit rates to code. The R-D performance for *Cheerleaders* D1 is presented in Fig. 13. The R-D performance for *Football* CIF is presented in Fig. 14. The same test with rate control being enabled is shown in Table IV.

It can be seen from the results, the motion vectors (in white), generated by the JM software for the *Football* sequence do not reflect true motions, as shown in Fig. 15. The motion vectors generated by CALM are closer to true motions, as shown in Fig. 16. This contributes to the reduction of bits used for coding the motion vectors.

When RDO is turned on, however, the R-D performance gain is limited, as shown in Table III. This is due to the exhaustive nature of RDO method where the real number of bits to code for each mode has been calculated. Therefore, the impact of the Lagrange multiplier as an estimation method on coding efficiency is not as significant any more.

V. Conclusion

We have proposed a simple and effective CALM method to improve the existing Lagrange cost function used for rate-constrained motion estimation. It was shown by analysis and experiments that the proposed method achieves better rate-distortion performance than the reference software, with negligible extra computational cost. The improvement is particularly significant for high motion videos. This method can be applied to both full search and fast motion estimation methods. This paper describes our work that led to the adoption by the Joint Video Team [30] and included in the reference software JM 12.0 onward [27].

References

[1] G. J. Sullivan and T. Wiegand, "Video compression: From concepts to the H.264/AVC standard," *Proc. IEEE*, vol. 93, no. 1, pp. 18–31, Jan. 2005.

[2] *Advanced Video Coding for Generic Audiovisual Services*, ITU-T and ISO/IEC JTC Rec. H.264 and ISO/IEC 14496-10 (MPEG-4) AVC, 2003.

[3] H. Everett, "Generalized Lagrange multiplier method for solving problems of optimum allocation of resources," *Oper. Res.*, vol. 11, no. 3, pp. 399–417, May–Jun. 1963.

[4] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 36, no. 9, pp. 1445–1453, Sep. 1988.

[5] P. A. Chou, T. Lookabaugh, and R. M. Gray, "Entropy-constrained vector quantization," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 37, no. 1, pp. 31–42, Jan. 1989.

[6] G. J. Sullivan and R. L. Baker, "Rate-distortion optimized motion compensation for video compression using fixed or variable size blocks," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, vol. 1. Phoenix, AZ, 1991, pp. 85–90.

[7] A. Ortega, K. Ramchandran, and M. Vetterli, "Optimal trellis-based buffered compression and fast approximations," *IEEE Trans. Image Process.*, vol. 3, no. 1, pp. 26–40, Jan. 1994.

[8] K. Ramchandran, A. Ortega, and M. Vetterli, "Bit allocation for dependent quantization with applications to multiresolution and MPEG video coders," *IEEE Trans. Image Processs.*, vol. 3, no. 5, pp. 533–545, Sep. 1994.

[9] T. Wiegand, M. Lightstonre, 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 standards," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 2, pp. 182–190, Apr. 1996.

[10] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Process. Mag.*, vol. 15, no. 6, pp. 74–90, Nov. 1998.

[11] A. Ortega and K. Ramchandran, "Rate-distortion methods for image and video compression: An overview," *IEEE Signal Process. Mag.*, vol. 15, no. 6, pp. 23–50, Nov. 1998.

[12] M. C. Chen and A. N. Willson, Jr., "Rate-distortion optimal motion estimation algorithms for motion-compensated transform video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 2, pp. 147–158, Apr. 1998.

[13] D. T. Hoang, P. M. Long, and J. S. Vitter, "Efficient cost measures for motion estimation at low bit rates," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 4, pp. 488–500, Aug. 1998.

[14] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. J. Sullivan, "Rate-constrained coder control and comparison of video coding standards," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 688–703, Jul. 2003.

[15] J. Sun, W. Gao, D. Zhao, and Q. Huang, "Statistical model, analysis and approximation of rate-distortion function in MPEG-4 FGS videos," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 4, pp. 535–539, Apr. 2006.

[16] Y.-K. Tu, J.-F. Yang, and M.-T. Sun, "Efficient rate-distortion estimation for H.264/AVC coders," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 5, pp. 600–611, May 2006.

[17] Q. Chen and Y. He, "A fast bits estimation method for rate-distortion optimization in H.264/AVC," in *Proc. Picture Coding Symp. (PCS)*, San Francisco, CA, Dec. 2004, pp. 133–134.

[18] C. S. Kannangara, I. E. G. Richardson, M. Bystrom, J. R. Solera, Y. Zhao, A. MacLennan, and R. Cooney, "Low-complexity skip prediction for H.264 through Lagrangian cost estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 2, pp. 202–208, Feb. 2006.

[19] L. Chen and I. Garbacea, "Adaptive estimation in Lagrangian rate-distortion optimization for video coding," in *Proc. SPIE Visual Commun. Image Process.*, vol. 6077(2). San Jose, CA, Jan. 2006, pp. 60772B.1–60772B.8.

[20] *Text Description of Joint Model Reference Encoding Methods and Decoding Concealment Methods*, document JVT-R095.doc, Joint Video Team of ISO/IEC JTC1/SC29/WG11 and ITU-T SG16/Q.6, Bangkok, Thailand, Jan. 2006.

[21] T. Wiegand and B. Girod, "Lagrange multiplier selection in hybrid video coder control," in *Proc. IEEE Int. Conf. Image Process.*, Thessaloniki, Greece, Oct. 2001, pp. 542–545.

[22] X. Yi, J. Zhang, N Ling, and W. Shang, *Improved and Simplified Fast Motion Estimation for JM*, document JVT-P021.doc, Joint Video Team of ISO/IEC MPEG and ITU-T VCEG, 16th Meeting, Poznan, Poland, Jul. 2005.

[23] X. Yi and N. Ling, "Improved normalized partial distortion search with dual-halfway-stop for rapid block motion estimation," *IEEE Trans. Multimedia*, vol. 9, no. 5, pp. 995–1003, Aug. 2007.

[24] J. Zhang, X. Yi, N. Ling, and W. Shang, "Bit rate distribution analysis for motion estimation in H.264," in *Proc. IEEE Int. Conf. Consumer Electron. (Dig. Tech. Papers)*, Las Vegas, NV, Jan. 2006, pp. 483–484.

[25] P. Sangi, J. Heikkila, and O. Silven, "Selection of the Lagrange multiplier for block-based motion estimation criteria," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, vol. 3. Montreal, Canada, May 2004, pp. 325–328.

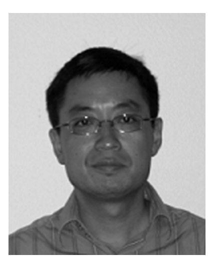
[26] K. Takagi, *Lagrange Multiplier and RD-Characteristics*, Joint Video Team of ISO/IEC MPEG and ITU-T VCEG, document JVT-C084.doc, 3rd Meeting, Fairfax, VA, May 2002.

[27] *JM Reference Software version 10.2* [Online]. Available: http://iphome.hhi.de/suehring/tml/download/

[28] G. Bjøntegaard, *Calculation of Average PSNR Difference Between RD-Curves*, document VCEG-M33.doc, ITU-T VCEG, 13th Meeting, Austin, TX, Apr. 2001.

[29] T. K. Tan, G. Sullinan, and T. Wedi, *Recommended Simulation Conditions for Coding Efficiency Experiments*, document VCEG-AA10.doc, ITU-T VCEG, Nice, France, Oct. 2005.

[30] J. Zang, X. Yi, N. Ling, and W. Shang, *Context Adaptive Lagrange Multiplier (CALM) for Motion Estimation in JM: Improvement*, document JVT-T046.doc, Joint Video Team of ISO/IEC MPEG and ITU-T VCEG, 20th Meeting, Klagenfurt, Austria, Jul. 2006.



Jun Zhang received the B.S. degree in computer science from the Huazhong University of Science and Technology, Wuhan, China, in 1988, and the M.S. degree in computer science from Nanjing University, Nanjing, China, in 1990. He is currently pursuing the Ph.D. degree in computer engineering from the Department of Computer Engineering, Santa Clara University, Santa Clara, CA.

From 2004 to 2007, he was a Senior Architect with EBay, Inc., San Jose, CA, before he founded Vercury, Inc., Pleasanton, CA, which specializes in video/audio fingerprinting technologies. His current research interests include motion estimation for video compression.

Xiaoquan Yi (S'03–M'05) received the B.A. degree in history from Hunan University of Science and Technology, Xiangtan, China, in 1993, and the M.S. and Ph.D. degrees from Santa Clara University, Santa Clara, CA, in 2001 and 2005, respectively, both in computer engineering.

He joined Google, Inc., Mountain View, CA, in 2005, where he is currently a Research Engineer. He is one of the key contributors for building Google/YouTube video transcoding infrastructure. His current research interests include video compression, communication, and content analysis.

Dr. Yi was a Session Chair at the IEEE International Symposium on Circuits and Systems, in 2007. He served as a Technical Program Committee Member of the IEEE International Conference on Consumer Electronics, in 2007, 2008, and 2009. His team's proposal on fast motion estimation (simplified and unified multi-hexagon search or simplified fast motion estimation) was adopted in 2005, and their proposal on Lagrange method (context adaptive Lagrange multiplier) was adopted in 2006, both into the H.264/MPEG-4 AVC video coding international standard.

Nam Ling (S'88–M'90–SM'99–F'08) received the B.Eng. degree in electrical engineering from the National University of Singapore, Singapore, in 1981, and the M.S. and Ph.D. degrees, both in computer engineering, from the University of Louisiana, Lafayette, in 1985 and 1989, respectively.

He has served as a Visiting Professor, Consultant, Scientist, and Scholar for many institutions, and as a Collaborator and Consultant to several companies. From 1981 to 1983, he was an Engineer with Hewlett Packard, Palo Alto, CA. He is currently a Full Professor of Computer Engineering and the Associate Dean for Research and Faculty Development with the School of Engineering, Santa Clara University (SCU), Santa Clara, CA. He has more than 140 publications in the fields of video coding and systolic arrays. He is the primary author of the book *Specification and Verification of Systolic Arrays* (World Scientific, 1999). His team's fast motion estimation (simplified and unified multi-hexagon search or simplified fast motion estimation) and Lagrange method (context adaptive lagrange multiplier) were both adopted into the H.264/MPEG-4 AVC video coding international standard and reference software. His current research interests include video coding.

Dr. Ling is a recipient of the Arthur Vining Davis Junior Faculty Fellowship, in 1991, the SCU Outstanding Achievement Award, in 1992, the SCU Engineering Researcher of the Year Award, in 1999, the SCU Award for Recent Achievement in Scholarship, in 2002, the SCU President's Recognition Award, in 2005, the SCU Award for Sustained Excellence in Scholarship, in 2007, and the SCU Engineering Award for Teaching Excellence, in 2010. He was named the IEEE Distinguished Lecturer for 2002–2003 and 2007–2008. He also received the 2003 IEEE International Conference on Consumer Electronics Best Paper Award (First Place Winner) for the work on MPEG-4 face animation. He is an IEEE Fellow due to his contributions to video coding algorithms and architectures. He served as a Keynote Speaker for the 2008 IEEE Asia Pacific Conference on Circuits and Systems, the International Workshop on Video Coding and Video Processing, in 2008, and the 2009 Joint Conferences on Pervasive Computing. He served as a Distinguished Invited Lecture Speaker for the IEEE Conference on Industrial Electronics and Applications, in 2010. He served as a Major Speaker and Panelist for several other conferences and seminars. He was the General Chair for the IEEE Hot Chips Symposium, in 1995, the General Co-Chair for the International Workshop on Video Coding and Video Processing, in 2008, and the Technical Program Co-Chair for the IEEE International Symposium on Circuits and Systems, in 2007, the IEEE Workshop on Signal Processing Systems, in 2000 and 2007, the International Workshop on Digital and Computational Video, in 2002, the IEEE Asia Pacific Conference on Circuits and Systems, in 2010, and the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, in 2010. He was a Track Co-Chair for the IEEE International Symposium on Circuits and Systems, in 2004–2006. He served as the Chair of the IEEE Computer Society Technical Committee (TC) on Microprocessors and Microcomputers, in 1993–1995 and the Chair of the IEEE Circuits and Systems Society Circuits and Systems for Communications TC, in 2006–2008. He is also a Member of two other IEEE TCs (Visual Signal Processing and Communications and Design and Implementation of Signal Processing Systems). He served as an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS I, in 2002 and 2003, and was a Guest Editor for the *Journal of Signal Processing Systems* (formerly the *Journal of VLSI Signal Processing Systems*) special issues, in 2006 and 2010. He also served in program committees, organizing committees, and as a Session Chair for many IEEE conferences. He also served on the editorial boards of several technical journals. He has delivered more than 100 invited, distinguished, keynote colloquia in nine different countries.

Weijia Shang (S'88–M'90) received the B.S. degree in electrical and computer engineering from the Changsha Institute of Technology, National University of Defense Technology, Changsha, China, in 1982, and the M.S. and Ph.D. degrees in computer science from Purdue University, West Lafayette, IN, in 1984 and 1990, respectively.

Before joining Santa Clara University, Santa Clara, CA, in 1994, where she is currently an Associate Professor and the Chair of the Department of Computer Engineering, she was an Assistant Professor with the University of Southwestern Louisiana, Lafayette, from 1990 to 1993. Her current research interests include parallel processing, computer architecture, algorithm theory, special-purpose very large scale integration bit-level processor array design, parallelizing compiler techniques, and nonlinear programming.

Dr. Shang received a scholarship for studying abroad from the Chinese Academy of Sciences, Beijing, China, and the Research Initiation Award and the Faculty Early Career Development Program Award from the U.S. National Science Foundation.