KSII TRANSACTIONS ON INTERNET AND INFORMATION SYSTEMS VOL. 5, NO. 8, August 2011
Copyright ⓒ 2011 KSII

Low-Complexity Motion Estimation for H.264/AVC Through Perceptual Video Coding

Byoungman An[1], Youngseop Kim[1] and Oh-Jin Kwon[2]
[1] Department of Electrical and Electronics Engineering, Dankook University
126 Jukjeon-dong, Suji-gu, Yongin-si, Gyeonggi-do, 448-701, Korea
[e-mail: {yesbman, wangcho}@dankook.ac.kr]
[2] Department of Electronics Engineering, Sejong University
98 Kunja-Dong, Kwangjin-Ku, Seoul 143-747, Korea
[e-mail: ojkwon@sejong.ac.kr]
*Corresponding author: Youngseop Kim

Received June 9, 2011; revised July 20, 2011; accepted August 15, 2011;
published August 29, 2011

Abstract

This paper presents a low-complexity algorithm for an H.264/AVC encoder. The proposed motion estimation scheme determines the best coding mode for a given macroblock (MB) by finding motion-blurred MBs; identifying, before motion estimation, an early selection of MBs; and hence saving processing time for these MBs. It has been observed that human vision is more sensitive to the movement of well-structured objects than to the movement of randomly structured objects. This study analyzed permissible perceptual distortions and assigned a larger inter-mode value to the regions that are perceptually less sensitive to human vision. Simulation results illustrate that the algorithm can reduce the computational complexity of motion estimation by up to 47.16% while maintaining high compression efficiency.

Keywords: Advanced video coding (AVC), motion estimation, video coding, high-efficiency video coding (HEVC)

This research was supported by the National Research Foundation fo Korea (NRF), the Korean government grant funded by the Korea government (MEST) [No. 2011-0010181, The Study of Green Video Compression and Joint Source and Channel Coding for Wireless Communication].

DOI: 10.3837/tiis.2011.08.005

1. Introduction

H.264/AVC is the newest international video coding standard [1]. It achieves higher compression efficiency than other standards [2] by using advanced coding techniques such as multiple-reference frame prediction and context-based adaptive binary arithmetic coding (CABAC) [3]. It enables the compression of video to 1.5–2 Mbps for standard-definition video and 6–8 Mbps for high-definition video by saving storage space, frequency spectrum, and channel bandwidth. Most of all, the variable block mode of H.264/AVC contributes to high compression efficiency; however, it also requires high computing power to determine the best compression mode. It is known that the motion-estimation process for determining the best mode occupies more than 60% of the computing time of the whole encoder [4]. Therefore, the development of fast and efficient motion estimation algorithms for H.264/AVC is a hot research topic [5][6][7][8][9][10][11][12][13][14][15].

Low-complexity motion-estimation algorithms for H.264/AVC were described in [5] and [6]. Fast inter-mode decision-making using the Lagrangian cost correlation proposed in [5] to determine the best coding mode for a given macroblock (MB) by estimating the rate distortion (RD) cost from the neighboring MBs in the previous frame. In [6], an early prediction algorithm was used to reduce complexity by estimating the Lagrangian RD cost function using an adaptive model for the Lagrangian multiplier parameter based on local-sequence statistics. In [7], the authors analyzed the quality fluctuations of different frame types and proposed a distributed RD control algorithm for high-definition video coding. The work reported in [8] used dynamic control of motion-estimation search parameters to reduce the complexity of an innovative motion-estimation system by configuring the number of reference frames, valid block modes, and the search area used to perform the motion-estimation task for a given MB. In [9], an RD performance-improved mode-decision method was proposed for H.264 intra-coding without changing the syntax and decoding process of H.264. This method uses inter-block mode-dependent decision criteria for the intra 4×4 mode of the original rate-distortion optimization (RDO) procedure. In [10], the authors proposed a weight-prediction process using the original pixels of the current block, instead of motion-compensated subpixels, to reduce the number of additional operations required. The authors of [11] proposed a system for managing the computational complexity of H.264/AVC video encoding in a real-time scenario. Their complexity management system controls the coding time of each frame of a video sequence with maintaining a high frame rate and avoiding significant frame quality loss.

On the other hand, taking human perception into consideration, the concept of perceptual image and video coding has been in existence for quite some time. The work described in [12] reviewed the physiological characteristics of human perception and addressed the most relevant aspects of video coding applications. Moreover, that work described the computational models and metrics which guide the design and implementation of a video coding system, as well as recent advances in perceptual video coding. In [13], an automatic distortion sensitivity analyzing process was proposed for bit allocation in rate-constrained video coding. The work reported in [14] used a rate control scheme at both the frame and MB levels. At the frame level, the algorithm estimated the target bits using a motion complexity measure which represented the amount of motion between two consecutive frames. In [15], a perceptual distortion masking measure was used for rate control, with the measure coupled into the RDO process. The success of this fast perceptual mode-decision algorithm depends heavily on accurate estimation of visual features, and therefore a more precise scheme needs to be developed. The contribution of this study is a method of managing complexity to reduce computational processing. The key concept of the proposed speed-dependent motion-estimation (SDME) algorithm is that, to save computation time, it assigns a larger intermode value to regions that are perceptually less sensitive to distortion than to more sensitive regions.

The remainder of this paper is organized as follows. Section 2 analyzes the problem of the H.264/AVC standard and describes the new low-complexity management approach in detail. Section 3 reports RD and execution-time results for the operation of the coding algorithm on both advanced video coding (AVC) test sequences and high-efficiency video coding (HEVC) test sequences [16][17]. The rate figures are calculated from actual compressed file sizes and on mean squared errors (MSE) or peak signal-to-noise ratios (PSNR) from the reconstructed videos as given by the algorithm. These results are put into perspective by comparison to JM 17.2, which is the original reference software version of MPEG and ITU-T. The conclusions of the paper are presented in Section 4.

2. Low-Complexity Motion Estimation

The proposed algorithm decreases computational processing by means of early identification of MBs. The proposed SDME model aims to reduce computation time by skipping certain MBs while maintaining RD performance. SDME is compared with a JM 17.2 encoder operating with the following parameters (hereafter called the “baseline encoder”):

1) baseline profile;
2) five reference frames, IPPP sequence type, CAVLC;
3) asymmetrical multi-hexagon search (UMHexagonS);
4) RDO mode selection enabled.

First, the motion-estimation method used in the JM encoder will be presented, and then the low-complexity motion estimation model will be described.

2.1 Motion Estimation in the JM Encoder

H.264/AVC provides five inter-modes and three intra-modes. The inter-modes include SKIP, 16×16, 16×8, 8×16, and P8×8 modes; the intra-modes consist of Intra 4×4, Intra 8×8, and Intra 16×16 modes. The inter-modes are used to estimate motion between frames; in addition, the P8×8 mode can be separated into 8×8, 8×4, 4×8, and 4×4 modes for high resolution. To estimate motion within one frame, the intra-mode is used, which uses nine directions to estimate the current pixel using neighboring pixels.

In contrast to previous video coding standards, H.264/AVC considers the distortion and the occurrence rate for selecting the best mode within a number of possible modes. The RDO contributes to determining the motion vector and choosing the best mode. It consists of the Lagrangian coefficients with their weighted distortion values and occurrence rates [18]. The RDO method calculates the RD cost for each MB mode. For this reason, the motion-estimation process has high computational complexity. To determine the best motion-estimation mode, the inter-mode first has to determine a motion vector and a reference frame in terms of a current frame. The motion vector and the reference frame are determined using Equation (1):

$$J_{motion}(MV, REF \mid \lambda_{motion}) = SAD(s, r(MV, REF)) + \lambda_{motion} \cdot R(MV, REF)\,. \tag{1}$$

where λ_{motion} is the Lagrangian coefficient which depends on the quantization coefficient, and $R(MV, REF)$ is the observed bitrate used to code the motion vector and the reference frame, which is chosen by means of a defined table. Equation (2) is used to calculate the sum of the absolute difference (SAD), where $SAD(s, r(MV, REF))$ is the sum of the absolute values of the differences between the original image and the motion-compensated image according to the motion-estimation algorithm:

$$SAD(s, r(MV, REF)) = \sum_{x\in H, y\in V}^{H,V} | s(x, y) - r(x - m_x, y - m_y) | . \tag{2}$$

where s is a pixel of an original block, r is a pixel of the corresponding reconstructed block, H and V are the width and height of the MB respectively, and (m_x, m_y) is the motion vector of the MB. In P8×8 mode, the encoder determines the optimized macro mode for each 8×8 block and the optimized direction in intra mode, along with the best mode, using the minimum cost as determined in Equation (3):

$$J_{\mathrm{mode}}(s, r, M \mid \lambda_{\mathrm{mod}e}) = SSD(s, r, M) + \lambda_{\mathrm{mode}} \cdot R(s, r, M) . \tag{3}$$

where λ_{mode} is the square of λ_{motion}. M is the MB mode; it is the predicted direction mode or sublevel of the MB. $R(s, r, M)$ is the observed bitrate in practice when the encoder codes the mode corresponding to M. Equation (4) describes the sum of squared differences (SSD), $SSD(s, r, M)$:

$$SSD(s, r, M) = \sum_{x\in H, y\in V}^{H,V} (s(x, y) - r(x - m_x, y - m_y))^2 . \tag{4}$$

The H.264/AVC rate-distortion algorithm calculates and determines the best mode in such a way that it occupies more than 60% of the computation time for the whole encoder. Using Equation (3), the encoder decides on the best mode and computes the actual bitrate, which requires a DCT / Hadamard transform, quantization / dequantization, an inverse DCT / inverse Hadamard transform, and entropy coding. The entropy coding uses Exp-golomb code, context-adaptive variable-length coding (CAVLC), and the CABAC; therefore, the novel fast mode-decision method is highly effective in reducing computational processing time. The following sections describe the speed-dependent mode-selection algorithm in detail and present the results of applying this algorithm within the framework of the H.264 test-mode encoder.

2.2 Human Visual System

The human visual system (HVS) plays a significant role in understanding perceptual video coding [19]. The eyes can track moving objects in a visual scene to keep the object of interest on the fovea and compensate for object motion to improve visual acuity, a phenomenon known as Smooth Pursuit Eye Movement (SPEM) [20]. Human beings cannot perceive fine-scale variations in visual signals because of the psychovisual properties of the HVS. Related psychophysical studies have demonstrated the level of contrast required to detect a flickering grating at different spatial and temporal frequencies [21]. The masking effect refers to the perceptibility of one signal in the presence of another signal in its spatial, temporal, or spectral vicinity [22]. Moreover, the sensitivity of the HVS relies on the background luminance and color of the stimuli. Visual sensitivity can be measured using a spatiotemporal contrast sensitivity function (CSF) [23].

2.3 ROI-based Video Coding

Region of interest (ROI)-based perceptual video coding has been in existence for quite some time. Earlier ROI-based video coding methods included restrictions on how the ROI could be defined. The foreground/background video coding method uses this approach [24]. Visual attention techniques have recently made it possible to identify multiple ROIs. **Fig. 1** shows the perceptual visual property of video coding sequences, in which the main observation to be made is that human vision is sensitive to movement of well-structured objects (MV_0 and MV_1) while tolerating large distortions in moving areas with random structures (MV_2 and MV_3). Note that the higher-speed objects have larger-magnitude motion vectors than the low-speed objects because the size of the motion vector is related to the distance between the current frame and the nearest reference frame.

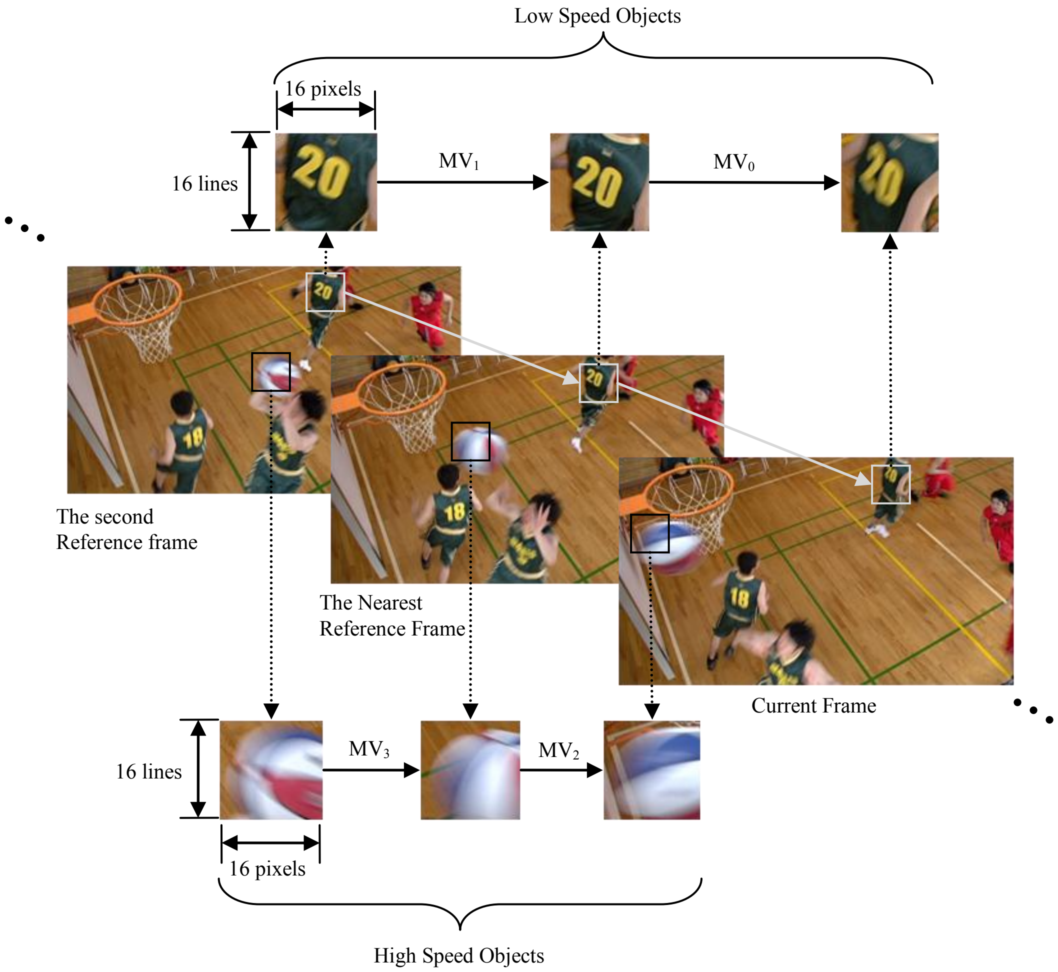


Fig. 1. Perceptual visual property of video coding sequences (WVGA : BasketballDrill sequences).

The theoretical key to the SDME algorithm is that it considers the speed of an object between the current frame and reference frames and codes more distant MBs less accurately when an object moves fast. This coding makes an attempt to control block distortion so that the perceptual impression given by different blocks will be consistent, i.e., blocks which can tolerate higher distortion will be more coarsely coded. Nevertheless, perceptual redundancy cannot be completely determined without consideration of the foveal features of the HVS. In this way, both computational savings and low complexity of the H.264/AVC encoding can be achieved.

2.4 Speed-Dependent Motion Estimation

To reduce computation time, the capability of the HVS to detect distortions in video sequences must be taken into account. As described previously, the basic idea of the proposed low-complexity motion-estimation scheme is to allocate fewer bits to areas where coding distortions are less noticeable. The HVS is more sensitive to perceptual distortions of smoothly textured objects with regular motions that are recognizable by eye movements in the foreground. These objects are generally susceptible to encoding distortions because of the inefficiency of the block-based motion-prediction model.

The SDME was designed based on these properties. First, to determine whether an MB has large movements and causes motion blur, the movement of a motion vector can be estimated as:

$$\mathrm{MVDistV} = |\overrightarrow{MV}| . \tag{5}$$

where MV is the motion vector used to code an MB between a current frame and a reference frame. It contains the direction of movement and the amount of motion. Next, let the distance value (DV) denote the maximum scale value as follows:

$$\mathrm{DV} = \sqrt{(width)^2 + (height)^2} . \tag{6}$$

where *width* and *height* are the horizontal and vertical size of the input image respectively. To calculate $MVDistV$ as a proportion of DV, these parameters are estimated using the offset value (OV), which can be obtained using Equation (7):

$$\mathrm{OV} = \frac{MVDistV}{DV} = \begin{cases} 1, 0 \le \mathrm{OV} \le 1 \\ 0, \text{otherwise} \end{cases} . \tag{7}$$

where OV is limited to the range $0 \le OV \le 1$.

Then, the lowest threshold (LT) and highest threshold (HT) were defined to judge whether or not the SDME is enabled. Equation (7) plays a significant role in implementing the SDME because the SDME algorithm is enabled only when OV is between LT and HT. The thresholds of LT and HT are the established range of input sequence before coding, minimum LT is 0 (0%) and maximum HT is 1 (100%), they are decided by considering the variety of experimental variables that are HVS and overall speed, frames, and search range of test sequences :

$$\text{Lowest Threshold}\,(LT) \le OV \le \text{Highest Threshold}\,(HT)\,. \tag{8}$$

To avoid reintroducing complexity into the coding process, the optimal choices for the *LT* and *HT* parameters are estimated before coding. To achieve computational savings, the *LT* and *HT* parameters are estimated using different *LT* and *HT* coefficients from 0.1% to 25% (**Table 1**). The appropriate values of *LT* and *HT* for the test sequence in order to reduce computational performance, BlowingBubble [QP (quantization parameter) = 28, frames = 50, search range = 64] were found to be *LT* ≈ 0.1% and *HT* ≈ 25%. However this result does not apply to other sequences because it occurs 0.59% additional bits along with high increased bits that can be a serious problem in terms of bandwidth and the communication channels. Through the experiments reported here, it was found that the threshold range between *LT* and *HT* is crucial because the SDME algorithm will not be beneficial and productive if the parameters are set too small or too large. In addition, through simulations considering the variety of experimental variables that are HVS and overall speed, frames, and search range of test sequences fixed parameters *LT* = 1.5 and *HT* = 25 were found to provide an acceptable tradeoff between computational performance and rate distortion. Therefore, these fixed parameters were chosen to provide low complexity with minimal loss of rate-distortion performance (see Section 3).

Table 1. The simulation results of proposed algorithm compared with that of the reference software, QP=28.

LT (%)	HT (%)	Original JM17.2				SDME				Original JM17.2 vs. SDME			
		Total time (sec)	ME time (sec)	PSNR (dB)	Compression ratio (%)	Total time (sec)	ME time (sec)	PSNR (dB)	Compression ratio (%)	Total time (%)	ME time (%)	△PSNR (dB)	△Compression ratio (%)
2	25	242.5	200.4	35.23	2.76	225.2	185.1	35.13	2.87	7.1	7.7	-0.1	+0.11
6	12	242.5	200.4	35.23	2.76	219.4	180.7	35.11	2.85	9.5	9.9	-0.12	+0.09
3	6	242.5	200.4	35.23	2.76	218.3	180	35.13	2.87	10	10.2	-0.1	+0.11
1.5	3	242.5	200.4	35.23	2.76	207.4	170.6	35.12	2.87	14.5	14.9	-0.11	+0.11
0.8	1.5	242.5	200.4	35.23	2.76	171.1	139.7	35.14	2.94	29.4	30.3	-0.09	+0.18
0.4	0.8	242.5	200.4	35.23	2.76	164	134	35.15	3.03	32.4	33.1	-0.08	+0.27
0.2	0.4	242.5	200.4	35.23	2.76	187.5	154	35.15	2.96	22.7	23.2	-0.08	+0.21
0.1	0.2	242.5	200.4	35.23	2.76	222	183.2	35.12	2.85	8.4	8.6	-0.11	+0.09
0.05	0.1	242.5	200.4	35.23	2.76	224	185	35.12	2.85	7.6	7.7	-0.11	+0.09
0.6	1.25	242.5	200.4	35.23	2.76	162.4	132.5	35.14	2.96	33	33.9	-0.09	+0.21
0.3	1.25	242.5	200.4	35.23	2.76	124.1	100.2	35.17	3.10	48.8	50	-0.06	+0.34
0.1	25	242.5	200.4	35.23	2.76	53.3	38.5	35.21	3.35	78	80.8	-0.02	+0.59

2.5 New Fast SDME Algorithm

One potential problem in performing mode prediction is error propagation due to a wrong prediction of the best mode and further prediction based on this nonoptimal mode. To avoid propagation of mode-prediction errors, a thorough mode-decision evaluation was carried out. The proposed SDME algorithm is shown in **Fig. 2**.

In the diagram, the motion vector and the reference frame are computed using Equations (1) and (2). *MVDistV*, *DV*, and *OV* are obtained when the proposed SDME algorithm is enabled, and then *OV* is compared with *LT* and *HT*. If *OV* is between *LT* and *HT*, the 16×16 mode is chosen as the best mode for calculating the MB without unnecessary computational processing, using Equations (3) and (4). The SDME algorithm plays a significant role in distinguishing between the standard method and the proposed scheme. By establishing the conditional process it enables the encoder to decide the best mode without recursive process of H.264/AVC.

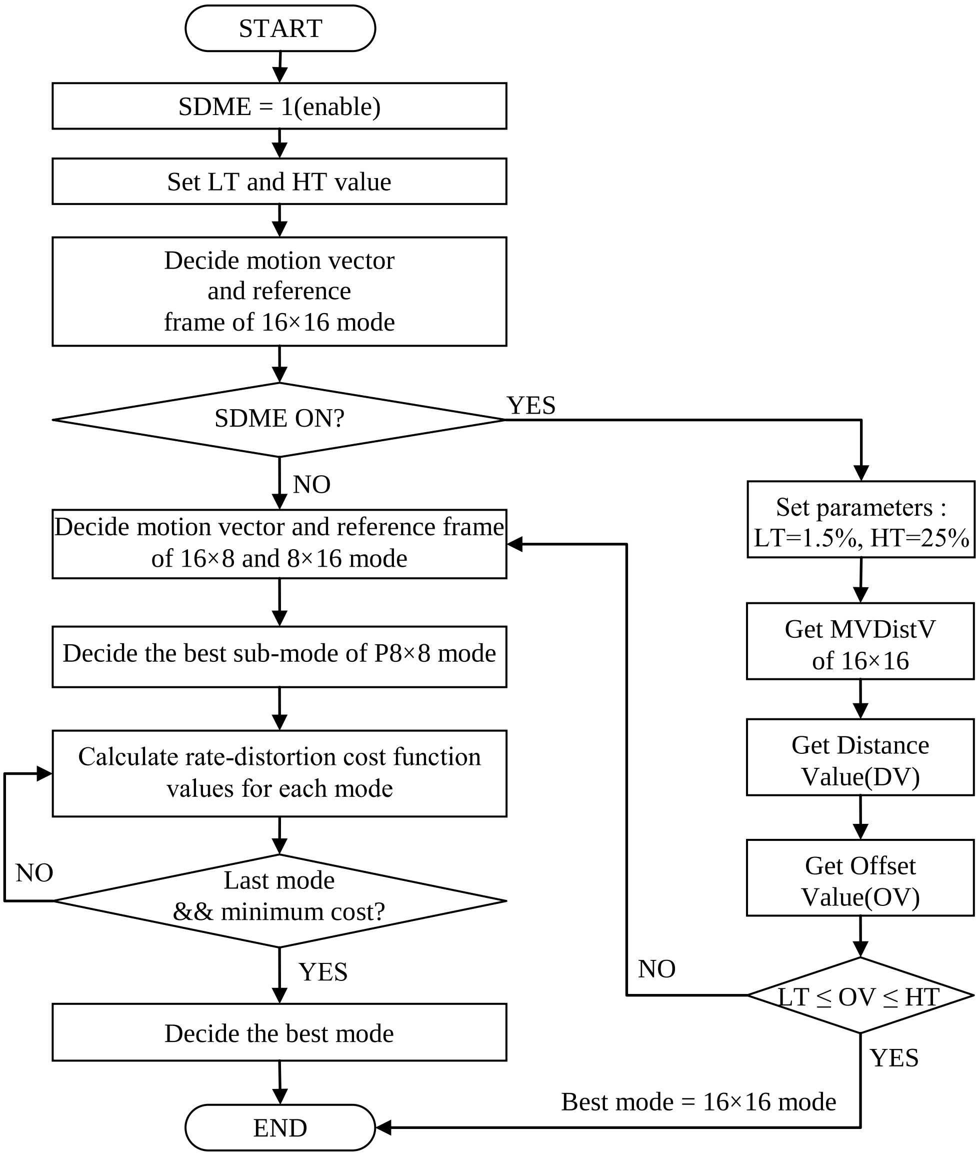


Fig. 2. Block diagram of the proposed SDME algorithm

3. Experimental Classification Results and Analysis

All simulations were implemented using Microsoft Visual Studio 2008 and a PC running Windows 7 Home Premium OS with an Intel ® Core™ I5 3.20-GHz CPU and 2 GB RAM. All experiments were performed by encoding and reconstructing an actual bitstream to verify the correctness of the proposed algorithm. It is important to observe that the bitrates are not entropy estimates; they were calculated from the actual size of the compressed files, and then the recovered image was compared with the original sequence. The distortion was measured using the PSNR.

Table 2 records the experimental results measured when the SDME algorithm is enabled, as well as the unmodified baseline profile encoder with respect to the total time, the Y-PSNR, and the compression ratio. The results shown in **Table 2** indicate that the coding time has been reduced by 11.52% to 47.85% compared with the baseline encoder due to the algorithm's early selection of the best mode. The computational savings are greater for larger-movement sequences such as Coastguard, Foreman, and Bqmall and lower for smaller-movement sequences such as News, BlowingBubble, and Paryscene.

Table 2. Coding results for H.264/AVC sequences and high efficiency video coding (HEVC) sequences, QP=28.

Input File		Search range	Frames	LT (%)	HT (%)	Original JM17.2			SDME			Original JM17.2 vs. SDME (%)		
Resolution	Seq. Name					Total time (sec)	PSNR (dB)	Compression ratio (%)	Total time (sec)	PSNR (dB)	Compression ratio (%)	Total Time (%)	△PSNR (dB)	△Compression ratio (%)
CIF (352*288)	Coastguard	64	30	1.5	25	3,393	35.43	3.40	1,792	35.22	3.62	47.16	-0.21	+0.22
	Foreman	64	30	1.5	25	2,507	36.83	1.35	1,356	36.72	1.64	45.91	-0.11	+0.29
	News	64	30	1.5	25	1,564	38.53	0.63	1,379	38.37	0.69	11.85	-0.16	+0.06
QCIF (176*144)	Coastguard	32	30	1.5	25	613	34.81	2.62	325	34.62	2.84	47.00	-0.19	+0.22
	Foreman	32	30	1.5	25	547	36.57	1.57	285	36.43	1.87	47.85	-0.14	+0.30
	News	32	30	1.5	25	371	37.12	0.89	327	37.01	0.93	11.64	-0.11	+0.04
WVGA (832*480)	Partyscene	64	50	0.1	25	5,265	34.80	4.06	1,903	34.65	4.65	63.86	-0.15	+0.59
	Partyscene	64	50	1.5	25	5,265	34.80	4.06	4,409	34.70	4.30	16.26	-0.1	+0.24
	Bqmall	64	60	1.5	25	4,334	36.70	1.81	2,531	36.63	2.16	41.58	-0.07	+0.35
	BasketballDrill	64	50	1.5	25	4,426	36.84	1.52	2,742	36.67	1.76	38.03	-0.17	+0.23
WQVGA (416*240)	BlowingBubble	64	50	1.5	25	1,212	34.97	2.31	1,072	34.87	2.46	11.52	-0.1	+0.15
	BQSquare	64	60	1.5	25	1,071	35.25	3.49	1,043	35.16	3.52	2.61	-0.09	+0.04

For the CIF and QCIF formats, Coastguard and Foreman show a time saving of approximately 45% with 0.12 dB PSNR degradation and 0.25% extra bits for Foreman, while News had the best RD efficiency and the worst time saving with 0.13 dB PNSR degradation, 0.04% additional bits, and a time saving of approximately 13%. For the HEVC test sequences such as WVGA and WQVGA format, Bqmall and BasketballDrill exhibited twice as much time saving as Partyscene and BlowingBubble, with percentage reductions of 41.58% and 38.03% respectively.

However, BQSquare, which was the smallest-movement sequence among the eight sample sequences, had the lowest time reduction, with 0.09 dB degradation, 0.04% extra bits, and a 2.61% time saving. Although the parameters for lower-speed sequences were set to provide an increased performance range (e.g., Partyscene with threshold values of $LT = 0.1$ and $HT = 25$), the computational time indeed decreased notably to half of its previous value, but the PSNR degradation and the number of additional bits dramatically increased to twice their previous values, with rates of 0.15 dB and 0.59% respectively. This is not desirable in applications with a constant-bitrate channel where perceptual quality should be maximized when processing power becomes limited as long as the bitrate limit is not exceeded.

However, it is worthwhile to evaluate the performance of the proposed SDME algorithm compared with that of the JM software with both bitrates and complexity set at comparable levels. This makes it possible to compare directly the original motion estimates and those from SDME for the perceptual qualities. The results for the Coastguard, Foreman, News, Partyscene, Bqmall, BasketballDrill, BlowingBubble, and BQSquare sequences with QP $\in \{25 \bullet\bullet\bullet 33\}$ are presented in **Fig. 3** and **Fig. 4**. In each case, the rate-distortion performance of the baseline profile and the low-complexity encoders is nearly identical across a range of test sequences and bitrates. Coding complexity and time are substantially reduced because RDO is selectively performed using the proposed algorithm. This reduction is particularly marked for high-activity sequences. In **Fig. 5**, the detailed features are subjective evaluation of the original JM and the proposed method. It indicates that HVS cannot subjectively notice more distortion in the proposed image than the original sequence and the sequence reconstructed with the original JM 17.2.

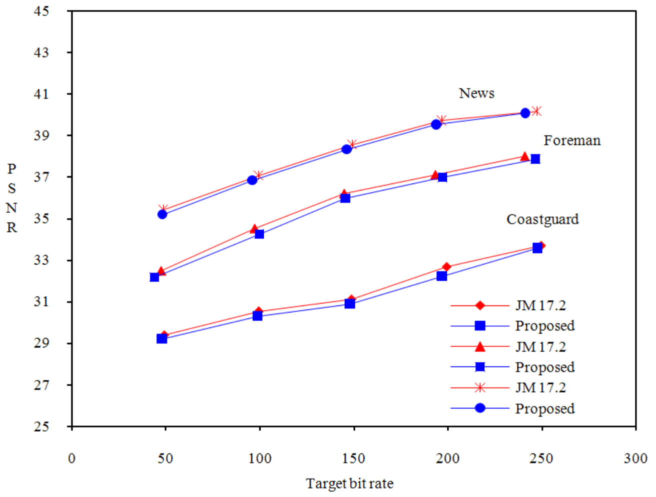


Fig. 3. Rate-distortion performance of QCIF sequences.

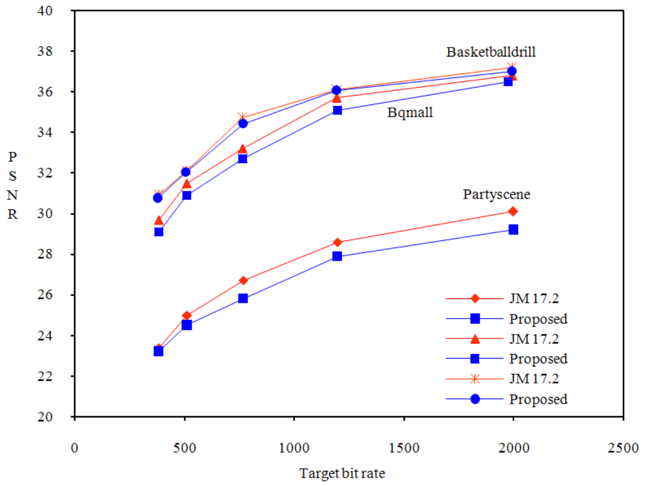
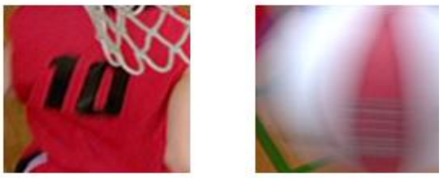
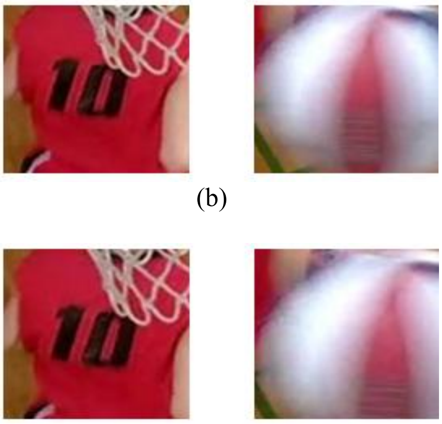


Fig. 4. Rate-distortion performance of WVGA sequences.



(a)



(b)

(c)

Fig. 5. Sub-images of BasketballDrill 120th frame, (a) the original sequence, (b) the sequence reconstructed with the original JM 17.2 software, (c) the sequence reconstructed with the proposed algorithm.

4. Conclusion

This paper has analyzed existing motion-estimation algorithms, as well as the HVS for perceptual video coding, and has presented the computation and performance tradeoffs involved in choosing reasonable values of *LT* and *HT*. This analysis directs the video coder to assign fewer bits to regions that can tolerate larger distortions, and accordingly computational savings are achieved. The proposed SDME algorithm is shown to be more effective than previous implementations of the JM17.2 reference software. The algorithm can reduce computational complexity by up to 47.16% with only about 0.22% extra bits and approximately 0.12 dB PSNR degradation. The authors believe that the results of this coding algorithm with HVS, along with its fast execution, are impressive.

References

[1] JVT-G050, “Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264/ISO/IEC 14496-10 AVC,” *Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG*, 2003. Article (CrossRef Link)

[2] Z. Chen, D. Zhang, K.N. Ngan, “An Efficient Algorithm for H.264/AVC High Definition Video Coding,” *IEEE trans. Consumer Electronics*, vol. 54, no. 4, pp. 1852-1857, Nov. 2008. Article (CrossRef Link)

[3] T. Wiegand, G.J. Sullivan, G. Bjontegaard, A. Luthra, “Overview of the H.264/AVC Video Coding Standard,” *IEEE trans. Circuits Syst. Video Technol*., vol. 13, no. 7, pp. 560-576, July 2003. Article (CrossRef Link)

[4] Y.J. Wang, C.C. Cheng, T.S. Chang, “A Fast Fractional Pel Motion Estimation Algorithm for H.264/MPEG-4 AVC,” *in Proc. ISCAS*, pp.3974-3977, May 2006. Article (CrossRef Link)

[5] S. H. Ri, Y. Vatis, and J. Ostermann, “Fast inter-mode decision in an H.264/AVC encoder using Mode and Lagrangian Cost Correlation,” *IEEE trans. Circuits Syst. Video Technol*., vol. 19, no. 2, pp. 302-306, Feb. 2009. Article (CrossRef Link)

[6] C.S. Kannangara, I.E.G. Richardson, M. Bystrom, J.R. Solera, Y. Zhao, A. Maclennan, 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. Article (CrossRef Link)

[7] Z. Chen, D. Zhang, K.N. Ngan, "An Efficient Algorithm for H.264/AVC High Definition Video Coding," *IEEE trans. Consumer Electronics*, vol. 54, no. 4, pp. 232-239, Nov. 2008. Article (CrossRef Link)

[8] S.S.M. Casula, F. Rovati, D. Alfonso, L. Fanucci, "Dynamic Control of Motion Estimation Search Parameters for Low Complex H.264 Video Coding," *IEEE trans. on Consumer Electronics*, vol. 52, no. 1, pp. 232-239, Feb. 2006. Article (CrossRef Link)

[9] J. You, C. Choi, J. Jeong, "Modified Rate Distortion Optimization using Inter-block Dependence for H.264/AVC Intra Coding," *IEEE trans. Consumer Electronics*, vol. 54, no. 3, pp. 1383-1388, Aug. 2008. Article (CrossRef Link)

[10] K. Panusopone, X. Fang, L. Wang, "An Efficient Implementation of Motion Estimation with Weight Prediction for ITU-T H.264 MPEG-4 AVC," *IEEE trans. Consumer Electronics*, vol. 53, no. 3, pp. 974-978, Aug. 2007. Article (CrossRef Link)

[11] C.S. Kannangara, I.E. Richardson, A.J. Miller, "Computational Complexity Management of a Real-time H.264/AVC Encoder," *IEEE trans. Circuits Syst. Video Technol*., vol. 18, no. 9, pp. 1191-1200, Sep. 2008. Article (CrossRef Link)

[12] Z. Chen, W. Lin, K.N. Ngan, "Perceptual Video Coding: Challenges and Approaches," in *Proc. of IEEE International Conference on Multimedia and Expo (ICME)*, pp.784-789, July 2010. Article (CrossRef Link)

[13] C.W. Tang, C.H. Chen, Y.H. Yu, C.J. Tsai, "Visual Sensitivity Guided Bit Allocation for Video Coding," *IEEE Transactions on Multimedia*, vol. 8, no. 1, pp. 11-18, Feb. 2006. Article (CrossRef Link)

[14] H. Yu, F. Pan, Z. Lin, Y. Sun, "A Perceptual Bit Allocation Scheme for H.264," in *Proc. of IEEE International Conference on Multimedia and Expo (ICME)*, July 2005. Article (CrossRef Link)

[15] C.J. Tsai, C.W. Tang, C.H. Chen, Y.H. Yu, "Adaptive Rate-distortion Optimization using Perceptual Hints," *IEEE International Conference on Multimedia and Expo*, vol. 1, pp. 667-670, 2004. Article (CrossRef Link)

[16] N11113, "Joint Call for Proposals on Video Compression Technology,*" Joint Collaborative Team on Video Coding (JCT-VC) of ISO/IEC MPEG and ITU-T VCEG, MPEG 91st meeting*, Kyoto, Japan, Jan. 2010. Article (CrossRef Link)

[17] JCTVC-B300, "Common Test Conditions and Software Reference Configurations," *Joint Collaborative Team on Video Coding (JCT-VC) of ISO/IEC MPEG and ITU-T VCEG, MPEG 93rd meeting*, Geneva, Jul. 2010. Article (CrossRef Link)

[18] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, G. Sullivan, "Rate-constrained Coder Control and Comparison of Video Coding Standard," *IEEE Trans. Syst. Video Technol.*, vol. 13, no. 7, pp. 688-703, Jul. 2003. Article (CrossRef Link)

[19] B. Wandell, "Foundations of Vision," *Sinauer Associates*, Sunderland, MA, 1995. Article (CrossRef Link)

[20] B. Girod, "Eye Movements and Coding of Video Sequences," in *Proc. of Visual Communications and Image Processing VCIP88*, SPIE vol. 1001, Cambridge, MA, USA, pp. 398-405, Nov. 1988. Article (CrossRef Link)

[21] D.A. Robinson, "The Mechanics of Human Smooth Pursuit Eye Movement," *J. Physiol.*, vol. 180, pp. 569-591, Oct. 1965. Article (CrossRef Link)

[22] N. Jayant, J. Johnston, R. Safranek, "Signal Compression based on Models of Human Perception," in *Proc. of IEEE*, vol. 81, no. 10, pp. 1385-1422, Oct. 1993. Article (CrossRef Link)

[23] D.H. Kelly, "Motion and vision II stabilized spatio-temporal surface," *J. Opt. Soc. Amer.*, vol. 69, no. 10, pp. 1340-1349, Oct. 1979. Article (CrossRef Link)

[24] D. Chai, K.N. Ngan, "Foreground/Background Video Coding Scheme," in *Proc. of IEEE International Symposium on*, vol. 2, pp. 1448-1451, 1997. Article (CrossRef Link)

Byoungman An received his B.S in Electronics & Electrical Engineering from Dankook University, Korea, in 2010. He is also currently pursuing an M.S degree in the school of Electrical & Electronics Engineering. His research interests include image/video compression and image processing.

Youngseop Kim received the M.S in Computer Engineering from the University of Southern California in 1991, and the Ph.D. in Electronic Systems from Rensselaer Polytechnic Institute in 2001. He was a manager at Samsung SDI until 2003. He developed the image-processing algorithm for PDP TV while at Samsung. Currently he is an Associate Professor at Dankook University in Korea. He is the resolution member and the Editor of JPsearch part 2 in JPEG, the co-Chair of JPXML in JPEG, and Head of Director (HOD) of Korea. He is also Editor-in-chief of the Korea Semiconductor and Technology Society. His research interests are in the areas of image/video compression, pattern recognition, communications, stereoscopic codecs, and augment reality. They include topics such as object-oriented methods for image/video coding, joint source-channel coding for robust video transmission, rate control, video transmission over packet wired or wireless networks, pattern recognition, and image processing.

Oh-Jin Kwon received a B.S. degree from Hanyang University, Seoul, Korea, in 1984, an M.S. degree from the University of Southern California, Los Angeles, 1991, and a Ph.D. degree from the University of Maryland, College Park, in 1994, all in electrical engineering. From 1984 to 1989, he was a research staff member at the Agency for Defense Development, Korea, and from 1995 to 1999, he was the head of the Media Laboratory at Samsung SDS, Seoul. Since 1999, he has been a faculty member at Sejong University, Seoul, Korea, where he is currently an Associate Professor. His research interests are image and video coding, watermarking, analyzing, and processing.