APSIPA ASC 2010

Asia-Pacific Signal and Information Processing Association Annual Summit and Conference

ENTER

December 14-17, 2010
Biopolis, Singapore

Organizer: A*STAR Institute for Infocomm Research

Supported by: Singapore Exhibition & Convention Bureau

December 16, Thursday (Day 2)
Matrix, Biopolis

TIME	PROGRAM					
0815 - 1700	Registration					
0900 - 1000	**Keynote 2** *Robots and the Human* Oussama Khatib *Chair : Haizhou Li* *Level 4, Breakthrough + Discovery*					
1000 - 1030	**Morning Tea Break**					
1030 - 1210	**Thu-AM. S5** **Advances in Digital Filters and Filter Banks**	**Thu-AM. S6** **Advances in Sparse Signal Processing**	**Thu-AM. R4** **Signal Processing Theory and Application**	**Thu-AM. R5** **Communication Systems**	**Thu-AM. PO3** **Speech, Language and Audio**	
	Chair : Mrityunjoy Chakraborty *Level 4, Breakthrough*	*Chairs : Andy W. H. Khong* *Level 4, Discovery*	*Chair : Julien Epps* *Level 4, Creation*	*Chair : Yoshikazu Miyanaga* *Level 4, Exploration*	*Chair : Norihide Kitaoka* *Level 4, Foyer*	
1210 - 1310	**Lunch Break** *Level 4, Foyer* **TC Meeting 4 :** Circuits, Design and Implementation *Level 3, Meeting Room 1* **TC Meeting 5 :** Speech, Language, and Audio *Level 3, Meeting Room 2* **TC Meeting 6 :** Image, Video, and Multimedia *Level 3, Meeting Room 4* Note: 3 TC Meetings cum lunch					**Thu-PM. ST** Signal Proces Theory and Application *Chair : Chan* *Chun Bao* *Level 4, Foy*
1310	**Forum Session** The Next 10 Years of APSIPA					
1330 - 1510		**Thu-PM. S7** **Recent Topics on Signal Processing for Active Noise Control**	**Thu-PM. R6** **Biomedical Signal Processing**	**Thu-PM. R7** **Virtual Reality and Media Processing**	**Thu-PM. PA2** **Recent Developments in Data Hiding and Forensics**	**Thu-PM. ST** Communicati and Circuits
	Chairs : Sadaoki Furui, H.-Y. Mark Liao & Lin-shan Lee *Level 3, Meeting Room 3*	*Chairs : Yoshinobu Kajikawa, Waleed H. Abdulla* *Level 4, Breakthrough*	*Chair : Cuntai Guan* *Level 4, Discovery*	*Chair : Weisi Lin* *Level 4, Creation*	*Chair : Jiwu Huang* *Level 4, Exploration*	*Chair : Peng* *Tan* *Level 4, Foy*
1510 - 1530	**Afternoon Tea Break**					
1530 - 1710	**Thu-PM. S8** **High-Efficiency Video Coding (HEVC)**	**Thu-PM. S9** **Medical Image Processing**	**Thu-PM. S10** **Recent Advances in Signal Representations - Filters, Transforms and Sparse Representations**	**Thu-PM. R8** **Circuits and Systems**	**Thu-PM. PO4** **Communication and Circuits**	
	Chairs : Chun-Jen Tsai, Wen-Hsiao Peng & Yo-Sung Ho *Level 4, Breakthrough*	*Chairs : Cuntai Guan, Jimmy Liu* *Level 4, Discovery*	*Chairs : Shogo Muramatsu, Akira Hirabayashi* *Level 4, Creation*	*Chair : Eliathamby Ambikairajah* *Level 4, Exploration*	*Chair : Kazunori Hayashi* *Level 4, Foyer*	
1800 - 2130	**Conference Banquet** @ Megu Hall, Singapore Flyer					

PAPER LIST

Thu-PM.S8

Thu-PM.S8.1
Towards the Next Video Standard: High Efficiency Video Coding
Hsueh-Ming Hang; Wen-Hsiao Peng; Chia-Hsin Chan; Chun-Chi Chen

Thu-PM.S8.2
Fast Motion Estimation with Extended Block Sizes for High-Definition Video Coding
Ji-Hee Moon; Yo-Sung Ho

Thu-PM.S8.3
High-Efficiency Video Compression Framework Based on Flexible Unit Representation
Sunil Lee; Woo-Jin Han; Jung-Hye Min; Il-Koo Kim; Elena Alshina; Alexander Alshin; Tammy Lee; Jianle Chen; Vadim Seregin; Yoon-Mi Hong; Min-Su Cheon; Nikolay Shlya
Ken McCann; Thomas Davies; Jeong-Hoon Park

Thu-PM.S8.4
Enhanced Intra Mode Dependent Directional Transform
Feng Zou; Oscar C. Au; Chao Pang; Jingjing Dai

Thu-PM.S8.5
Parametric Overlapped Block Motion Compensation for Pixel-Adaptive Temporal Prediction
Yi-Wen Chen; Chung-Lin Lee; Wen-Hsiao Peng

Fast Motion Estimation with Extended Block Sizes for High-definition Video Coding

Ji-Hee Moon and Yo-Sung Ho

Gwangju Institute of Science and Technology (GIST)
1 Oryong-dong, Buk-gu, Gwangju, 500-712, Korea
E-mail: {jhmoon, hoyo}@gist.ac.kr Tel: +82-62-715-2263

***Abstract*—H.264/AVC video coding standard performs motion estimation using various motion partition sizes. It has been shown to improve the coding gain. However, video coding using blocks bigger than 16×16 pixels can provide a substantial coding gain compared to H.264/AVC, especially when applied to high definition video sequences. Even if coding efficiency is improved, the complexity of motion estimation using extended motion partition sizes is also increased. For high definition video coding, fast motion estimation using extended motion partition sizes has been proposed. In order to reduce the complexity, we propose a fast motion estimation method for extended block sizes. Experimental results show that the proposed algorithm reduces the complexity of motion estimation without any visible performance degradation.**

I. INTRODUCTION

H.264/AVC is the latest video coding standard that provides high compression efficiency [1]. For higher compression efficiency, H.264/AVC has adapted several useful coding techniques, such as variable block-size macroblock modes, multiple reference frames, integer discrete cosine transform (DCT), and efficient entropy coding techniques. However, it cannot provide the sufficient performance in the high resolution video sequences.

The joint collaborative team on video coding (JCT-VC) has been established to develop the high efficiency video coding (HEVC) standard [2]. The purpose of this work is to provide higher compression capability than the H.264/AVC standard.

In order to achieve higher coding efficiency in high-definition contents, video coding using motion partitions larger than 16 × 16 has been proposed. It has been demonstrated that video coding using blocks bigger than 16 ×16 pixels can provide substantial coding gain compared to H.264/AVC, especially when applied to high resolution video sequences. The correlation between neighboring pixels is increased in high resolution video contents. Therefore, When we apply the video coding using blocks bigger than 16 ×16 pixels to high-definition video sequence, the residual error can be reduced effectively.

The improved inter prediction methods such as quarter-pixel precision and various macroblock modes make H.264/AVC to estimate the motion information in a sequence accurately. In H.264/AVC, motion estimation takes approximately 50~60% of a total encoding time. In order to reduce the complexity of motion estimation, several researches about fast motion estimation have been proposed.

When we apply larger block sizes to motion estimation, the complexity of video encoder is increased. There are lots of integer pixel values to be calculated for motion estimation. It is disadvantage of the video coding using larger block sizes. Thus, enhanced predictive zonal search for high-definition video coding is recommended.

In this paper, we propose a fast motion estimation method which adopts an adaptive search range for video coding using large block. A search range is determined by using spatial and temporal correlation of neighboring motion vectors (MVs) of the current macroblock. We consider the distance between the current macroblock and the reference frame to determine the final search range. In order to enhance the accuracy of the starting point of search, we consider the relationship between the predictive motion vector and previously encoded motion vector.

II. THE CONVENTIONAL METHOD

A. *H.264/AVC Full Search Algorithm*

In order to predict the motion vector of the current macroblock, we need to check the spatial correlation between neighboring blocks. The predictive motion vector is determined by a median value of each neighboring horizontal motion vector and vertical motion vector [3].

Figure 1 shows the full search algorithm of H.264/AVC.

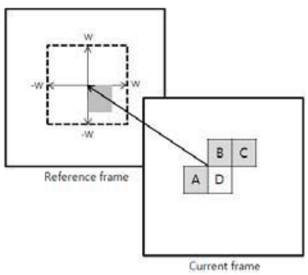

Fig. 1. H.264/AVC full search algorithm

In motion estimation, the starting point of search is determined according to the predictive motion vector. Then,

Proceedings of the Second APSIPA Annual Summit and Conference, pages 619–622,
Biopolis, Singapore, 14-17 December 2010.

the best matching block within the square search range is found. If the initial search range is w, the total search points are $(2w + 1)^2$ in the integer pixel level.

After the best matching position is determined in the integer pixel level, sub-pel motion estimation find the best matching position in half pixel level and quarter pixel level. The three step search is applied to sub-pel motion estimation.

B. Macroblock of extended block size

In addition to the existing motion partition sizes (16×16, 16×8, 8×16, 8×8, 8×4, 4×8, 4×4) in H.264/AVC, inter coding using 32×32, 32×16, and 16×32 partitions is also enabled in key technology area (KTA). Figure 2 shows the motion block partitions implemented in KTA software.

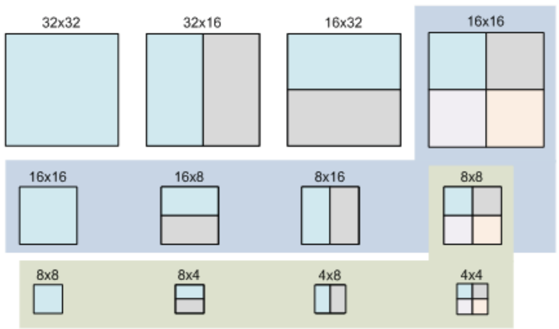

Fig. 2. Motion partitions used in the 32×32 blocks.

For each 32×32 block, mb32_skip_flag encodes the SKIP or DIRECT mode in the way similar to that of H.264/AVC. In additions, the original mb_type for M×N (M=8, or 16 and N=8, or 16) partition in H.264/AVC is also used to signal 2M×2N partition in a 32×32 block. For example mb_type equal to 3 for 8×16 macroblock partition in a p slice of H.264/AVC is also used for 16×32 partition in a P slice of KTA. The mb32_type is binarized in a similar way as mb_type of H.264/AVC and shares the same context index as mb_type of H.264/AVC [4].

If partition bigger than 16×16 is used, one motion vector per prediction list is signaled for each partition. Residual coefficients are signaled in a hierarchical way. A 1 bit syntax cbp32 has been introduced for the whole 32×32 block with 1 meaning that the whole block has at least 1 nonzero coefficient and 0 meaning that there is no coefficient with a magnitude greater than zero.

If the mb32_type of a 32×32 indicates that a 16×16 partition is used, then the four 16×16 blocks are signaled in raster scan order by using the same syntax elements as macroblock_layer function in H.264/AVC reference software except that the delta QP information is sent once for the whole 32×32 block. In other words, the whole 32×32 block shares the same QP. Each 16×16 block may be partitioned further in the quadtree manner, from size 16×16 down to size 4×4. Also the signaling of cbp and residual coefficients remains the same as in H.264/AVC. For macroblock size 64×64, each 32×32 block will be handled in the same way described above case of macroblock size 32×32.

III. PROPOSED ALGORITHM

It has been observed that the motion vector of the current marocblock is the same or very close to the motion vectors of its spatially and temporally adjacent blocks [5]. Considering motion vector correlation, we determine a new search range. Also, we set the starting search point of motion estimation based on the predictive motion vector and previously encoded motion vector.

A. Method using Spatial Correlation

Since the conventional motion estimation computes the all points in search range, the computational cost is high. In order to reduce the computational cost, we decrease a search range by considering the variance of the motion vectors of left, above, and above and to the right of the current macroblock. We can determine whether motion of the current macroblock is large or not by considering the variance of neighboring motion vectors. For example, if the variance of the horizontal motion vectors is larger than that of the vertical motion vectors, it means that the horizontal motion is larger than the vertical motion.

Since we assign the search range smaller than initial search range to the direction of small motion, the determined search range has different size about the horizontal direction and the vertical direction. When the calculated search range is larger than the initial search range, we do not apply the proposed algorithm.

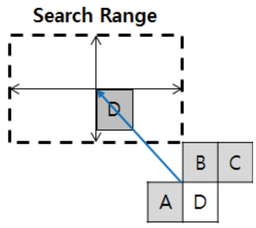

Fig.3. Adaptive search range.

Figure 3 illustrates an example of an adaptive search range. When the search range is reduced, it is possible to remove the best matched point. In order to guarantee coding efficiency of motion estimation, we apply a nonlinear function. To avoid that the determined search range is larger than the initial search range, the search range is modified by scaling factor α. The determined search range using spatial correlation is defined as follows:

$$R=max(1,\ \alpha\sqrt{\text{Standard deviation(MVs)}}\ ) \qquad \alpha=search\_range/2\sqrt{\text{Search\_range}} \tag{1}$$

where R is the determined search range based on the spatial correlation using the standard deviation of neighboring motion vectors and MV_s is neighboring motion vector of the current macroblock.

B. Method using Temporal Correlation

The determined search range is applied over all multiple reference frames. However, we can see that the nearest reference frame is often selected as the best reference frame for the current macroblock in Fig. 4.

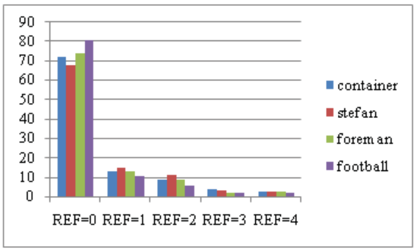

Fig. 4. Probability of chosen reference frame.

In order to reduce computational time of motion estimation far from the current macroblock, we reduce the search range again in further reference frame. Figure 5 illustrates the reduced search range of further reference frame.

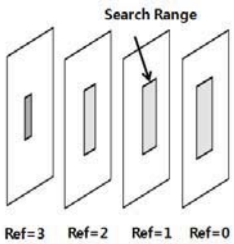

Fig. 5. Reduced search range for multiple reference frames.

First, we calculate the difference between the predictive and the best motion vectors of reference frames. The small difference means the best motion vector close to initial point and the large difference means the best motion vector far from initial point. Second, the search ranges of further reference frames are set with the doubled median value among differences. If the doubled median value is larger than the initial search range, this process is ignored. The determined search range based on the temporal correlation is defined as follows:

$$MVD_{ref} = | Best\_MV_{ref} - Predictive\ MV_{ref} |$$
$$r = median\{ MVD_{ref} \} \quad (2)$$
$$0 \le ref \le 3$$

C. Final search range decision

We consider the spatial correlation and the temporal correlation simultaneously to determine the final search range.

If the reference frame is farther apart from the current macroblock, the weight of search range based on the temporal correlation is increased. The weight depends on the distance between the current macroblock and the corresponding reference frame index. The final search range decision is decided by Eq. (3).

$$Final\_SR = (R + (Distance \times r))/(Distance + 1) \quad (3)$$

where *Distance* is the interval between the reference frame and the current macroblock. *R* is the search range based on the spatial correlation and *r* is the search range based on the temporal correlation.

After we apply the final search range to motion estimation process, the search range in further reference frame is reduced. Therefore, we can reduce the search points to be checked.

D. Adaptive starting search point decision

In this paper, we propose the method that reduces the search points based on the spatial correlation and the temporal correlation. Therefore, the accuracy of the proposed motion estimation can be decreased. In order to enhance the accuracy of motion estimation, we propose an adaptive starting search point decision. If we find starting search point close to motion vector which is determined using motion estimation process, the accuracy of the proposed motion estimation is enhanced. The proposed starting search point is defined as follows:

$$Determined\_pmv_{ref} = (pmv + mv_{ref-1})/2$$
$$1 \le ref \le 3 \quad (4)$$

where *pmv* is the conventional predictive motion vector and mv_{ref} is previously encoded motion vector. A new starting search point is average of the predictive motion vector and the estimated motion vector. It traces the motion of contents between successive frames. Therefore, we can improve the accuracy of the proposed motion estimation.

IV. EXPERIMENTAL RESULTS

The proposed fast motion estimation method for extended block sizes is implemented into KTA 2.7 software [6]. The detailed encoding parameters for the experiment are summarized in Table 1. PC environment for whole test is Windows XP, Six-Core AMD Opteron(tm) processor 8435 2.59GHz @ 3.73GB RAM.

TABLE I
ENCODING PARAMETERS

Profile	100 (high profile)
QP	22, 27, 32, 37
SearchRange	128
SymbolMode	1 (CABAC)
FrameStructure	IPPP…P
UseIntraMDDT	1
UseExtMB	2 (64×64)
Transform8×8mode	1
SearchMode	Full search
NumberReferenceframes	4

In order to examine the performance of the proposed fast motion estimation, we performed experiments on several test video sequences of the YUV 4:2:0 formats with 720p resolution (1280×720).

To evaluate our proposed algorithm, processing time comparison result is shown in Table 2.

TABLE Ⅱ
PROCESSING TIME SAVING COMPARISON (%)

Sequence	QP	Original	Proposed	△Time (%)
Jets	22	26773.16	12362.24	-53.83
	27	18628.08	7857.58	-57.82
	32	12204.86	5144.02	-57.85
	37	8391.66	3896.35	-53.57
Crew	22	29600.69	13037.08	-55.96
	27	26040.99	99736.00	-61.70
	32	21645.74	81258.28	-62.46
	37	16576.37	63817.24	-61.50
City	22	24219.14	13515.19	-44.20
	27	23706.94	11445.61	-51.72
	32	23326.60	9475.93	-59.38
	37	22010.32	7856.75	-64.30
Night	22	22628.35	12321.41	-45.55
	27	21322.94	10527.71	-50.63
	32	19614.12	88564.16	-54.85
	37	17359.79	73733.32	-57.53
Average				-55.80

In Table 2, we can notice that the proposed algorithm shows better results than the original full search method in the KTA software. Averagely, our proposed algorithm obtained 55.80% more time reduction than the original full search method in the KTA software.

TABLE Ⅲ
PSNR AND BIT RATE REDUCTION (%)

Sequence	QP	△Br	△ PSNR
Jets	22	1.50	0.03
	27	-0.17	0.02
	32	1.70	0.00
	37	-0.00	-0.06
Crew	22	-0.93	-0.01
	27	-0.37	0.02
	32	1.15	-0.05
	37	0.71	-0.10
City	22	1.46	0.03
	27	0.52	-0.01
	32	0.25	-0.03
	37	0.68	0.02
Night	22	-0.04	-0.01
	27	-0.31	-0.02
	32	0.80	-0.01
	37	1.46	-0.01
Average		0.53	-0.01

Table 3 shows the PSNR and bit rate difference. Note that the positive value means increment and negative value means decrement. From Table 3, we can know that the proposed algorithm does not have significant distortion and bit rate increased compared with KTA software.

Figure 6 demonstrates the rate distortion curves. As shown Fig. 6, there is no significant degradation of PSNR and bit rate.

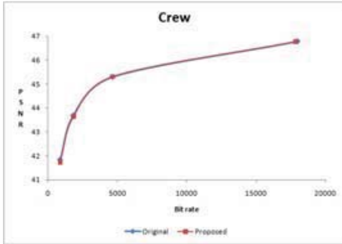

Fig. 6. Rate distortion curve.

V. CONCLUSION

We have proposed an adaptive search range decision method based on the spatial and the temporal correlation for the video coding using extended block sizes. We consider the variance of neighboring motion vectors and the distance between the current macroblock and the reference frames. The final search range is determined using weighted average. In addition, we proposed an adaptive starting search point decision to improve the accuracy of motion estimation. It is possible to integrate the proposed algorithm into several well-known fast search methods. We can reduce motion estimation time by 55.80% without significant degradation.

ACKNOWLEDGMENT

This research was supported by the MKE(The Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the NIPA(National IT Industry Promotion Agency) (NIPA-2010-(C1090-1011-0003))

REFERENCES

[1] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard, " IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, no. 7, pp. 560–576, July 2003.

[2] Joint Call for Proposals on Video Compression Technology, ISO/IEC MPEG 91st Meeting, Kyoto, Japan, Document N11113, April 2010.

[3] I. Richardson, *H.264 and MPEG-4 Video Compression*, Wiley, 2003, pp. 175–177.

[4] K. Lee et al., "Technical consideration for Ad Hoc Group on New Challenges in Video Coding Standardization," ISO/IEC JTC/SC29/WG11 Document M1558, July 2008, Hannover, Germany

[5] J. Chalidabhongse and C.-C. Kuo, "Fast motion vector estimation using multiresolution-spatio-temporal correlations," IEEE Transactions on Circuits and Systems for Video Technology, vol. 7, no. 3, pp. 477–488, June 1997.

[6] KTA Software Version 2.7, available at: http://iphome.hhi.de//suehring/tml/download/KTA/jm11.0kta2.7.