# Using the Predictive Method of Adaptive 4x4/8x8 Transformations to Reduce the Computation in Motion Estimation

K.-H. Chou and C.-P. Chung
Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan 300, ROC
Email: kwjc3348.cs96g@g2.nctu.edu.tw, cpchung@cs.nctu.edu.tw

***Abstract*—In order to improve coding compression and display quality in bitstream playback, an H.264, with an "adaptive transform with variable block" (ATVB), is able to achieve this. However, these added compression and display benefits are offset by the demand for double the computation power requirements. In addition to this computing power increase an extra round of Motion Estimation (ME), when using 4x4 or 8x8 a transform module is also required. Therefore, this paper proposes a savvy hypothesis to reduce the computation requirements between 30%-45% using ATVB with 0.25db display distortion and 2.5% bitrate increment.**

***Index Terms*—adaptive transformation, motion estimation, computation reduction, rate-distortion optimization**

## I. Introduction

H.264/MPEG-4 AVC [1] is the standard of video decoders that was specified by the ITU-T Video Coding Experts Group (VCEG) together with the ISO/IEC Moving Picture Experts Group (MPEG) for achieving higher compression and better display quality than previous codecs. Although, the video compression and display quality are the primary targets of the modern codec, the accompanying computations still need to be involved into design considerations.

In a modern encoder, Motion Estimation (ME) provides a great improvement in bitrate compression. It uses different sizes of blocks and their corresponding Motion Vector (MV) in place of intermediates residuals for reducing the coding samples. If the encoder is not constrained by any conditions, the bully-force searching scheme of ME would be technically intuitive enough to implement the encoder. However, bully-force searching ME is not a rational approach to apply to some video encoding devices, such as portable devices and embedded systems, based on the factors of power consumption and storage requirements.

The biggest proportion of encoding time is consumed by ME (Fig. 1), even though it has to be restricted to within a 32x32 search window and a limited number of reference frames. A number of papers have been written over the years contributing ideas as to the most selective ME method. [2]-[14] The majority of these papers paid attention to the balance between computations and bitrates within the viewpoint of the ME module.

However, each module within H.264 encoding are highly relative, so this paper will stand on the position between ME and Discrete Cosine Transform (DCT) and delineate another encoding method to accomplish the purpose of computational reduction.

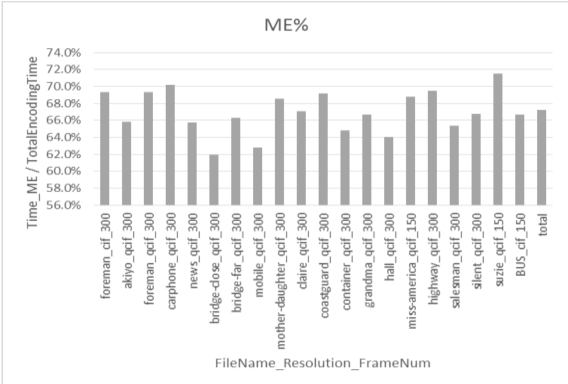


Figure 1. The execution percentates for ME. In the figure, the highest percentage of encoding time used for ME is "suzie_qcif_150".

(Note 1: encoded YUV files are denoted as the concatanation of file names, resolutions, and frame numbers.)

(Note 2: The statistic is the result with B frame disalbed)

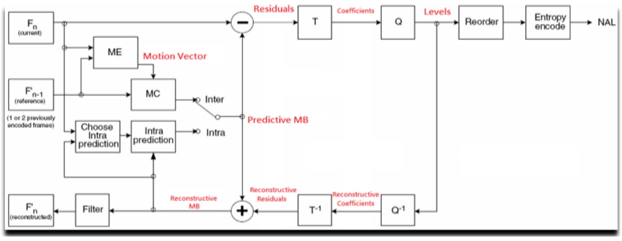


Figure 2. The flowchat of H.264 encoding.

Now, let's briefly recap the correlation between ME and DCT that H.264 (Fig. 2), has engaged in:

- *Fn* is the frame of current encoding. And each frame is operated in the unit of 16x16 luma macroblock (MB).
- In ME, current *Fn* is compared with reference frame(s), as reference *F'n-1*. An inter-prediction function, i.e. ME, searches similar corresponding values of MB of *F'n-1* that compare with current

Manuscript received June 24, 2014; revised October 21, 2014.


doi: 10.12720/ijsps.3.2.112-117

MB of *Fn* within a predefined region (such as 32x32 search window) according to seven optional partitions and Motion Vector (MV). The MV is the offset(s) between the position of the block of *Fn* and then determines the correct reference block of *F'n-1*.

- At the same time, an intra-prediction function is formed.
- The predictive MB *P* can be the intermediate of ME or intra-prediction using decision criteria (such as SA(T)D, SSD, or RDO).
- Residual *D* is the difference between current MB and *P*; *D* can then be transformed by the block size of 8x8 or 4x4.
- The Level of *X* is the transformed coefficient.
- The reconstructive MB is generated from the inverse encoding method.
- Subsequently, we will discuss the details about ME, DCT, and ATVB in the Background section, including observations. The hypothesis was inspired by these observations. Some experimental statistics were used to support the feasibility and worth of this paper's proposal. Finally, the conclusion will give the agenda for future works.

## II. Background

In order to achieve higher compressive rates as shown in Fig. 3, the ATVB has been the default setting in previous JM18.4 software. Although further compression can alleviate the request of storage size or network bandwidth, increased computation requirements are needed for the adaptive transform block decision of each block (that are shown as Fig. 4). From Fig. 1, it has shown that the most of the computation comes from Motion Estimation (ME). Unfortunately, ATVB needs a second processing series to calculate the ME and DCT in order to achieve the correct bitrate reduction. Hence, this paper will introduce an encoding manner that can reduce most unnecessary computations while ATVB is enabled

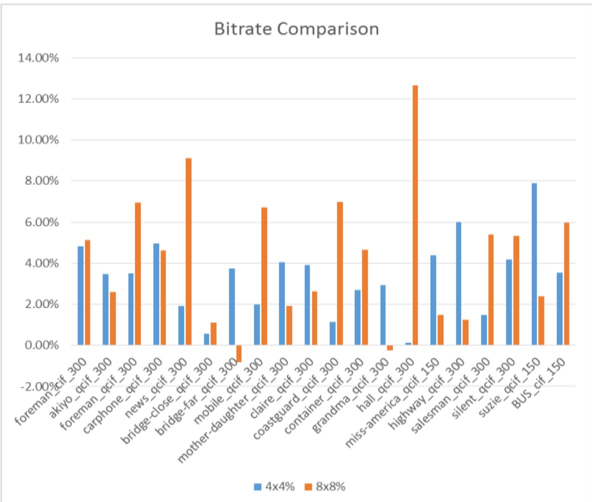


Figure 3. Bitrate decrement of ATVB. Taking "hall_qcif_300" as example, there is rough 13% increment of bitrate between ATVB and 8x8 DCT. But there is almost the same bitrate between ATVB and 4x4 DCT.

(The formula is that, decre_bitrate = (Bitrate_XxX / Bitrate_ATVB) – 1)

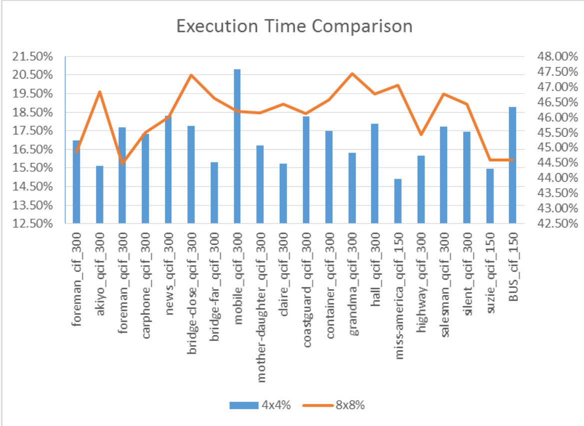


Figure 4. Execution time increment of ATVB. On average, ATVB increases about 45% with 8x8 DCT and about 18% with 4x4 DCT

(The formula is that, *incre_execution = (1- (Execution_XxX / Execution_ATVB))*

(Note: the left scale in y-axis is used for 4x4; the right scale in y-axis is used for 8x8)

Rate Distortion Optimization (RDO) is one of H.264 criterion to select the encoding decision and its result to obtain the best balance between bitrate and display quality. When the RDO is selected as the criteria to the encoding decision, the additional computation that was used for ATVB may not be worth the slight increase in improvement in display quality (shown as Fig. 5).

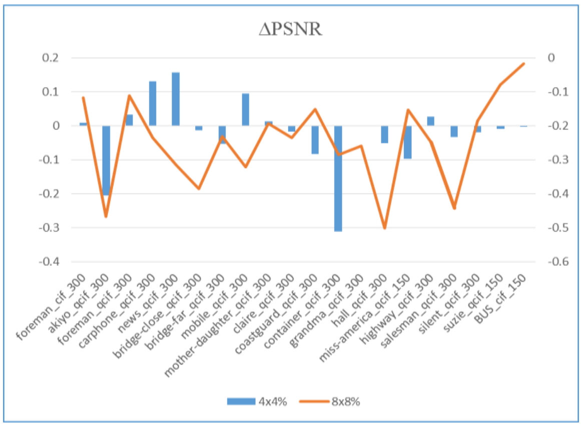


Figure 5. PSNR difference between 4x4 DCT or 8x8 DCT with ATVB in High-complex setting of encoder configuration.

(The formula is that, *diff_PSNR = (PSNR_XxX - PSNR_ATVB))*

The original goal of ATVB was designed to select better visual quality by using the criteria of SA(T)D, SSD [15] (when the RDOptimization encoder configuration is set at "0: RD-off (Low complexity mode)"). When using the RDO [15] tool as shown in Fig. 5, the ATVB determines the optimum block size ((4x4), (8x8)) to be selected to best optimize the balance of bitrate and display quality (when RDOptimization of encoder configuration is set as "1: RD-on (High complexity mode)"). Only some videos shown in Fig. 5 gain the better PSNR by encoding with 4x4 DCT. Because RDO is enabled in this experiment, and display quality isn't the sole target in the RDO's concerned. In RDO equation, the factor $\lambda_{MODE}$ can be altered to bias toward display quality or bitrate. The RDO formula is list as (1)

$$RDO\text{-}cost = Distortion + \lambda_{MODE} * Rate \qquad (1)$$

Originally, ATVB was invented to combine the advantage when DCT can dynamically select the better block-size for the good display quality. Theoretically, the 8x8 block size DCT is possible to contribute higher compression ratio than the one that using the 4x4 block size DCT based on the same quantization parameter (QP). Contrarily, the 8x8 block size DCT may generate worse display quality than that using the 4x4 block size DCT according to the identical QP.

On the side of ATVB's input, ATVB is highly correlative to its previous encoding module, Motion Estimation (ME). If the predictive block is completely match the current block, it means that ME find the identical predictive block with current encoding block, so that 8x8-block DCT is the pertinent decision for small bitrate.

On the side of ATVB's output, ATVB is also highly correlative to the quantization. If the *predictive block* far mismatches with the current block, it means that ME didn't find the similar predictive block with *current block* within the search window, hence, 4x4 DCT is the appropriate decision because it can decrease the possibility of display quality loss after quantization.

## III. OBSERRVATION

Firstly, the block size of DCT is must smaller than the block size of ME. It means that 8x8-block transform doesn't be used when the size of prediction blocks are smaller than 8x8 block. The block's size and its partition is depicted as below Fig. 6:

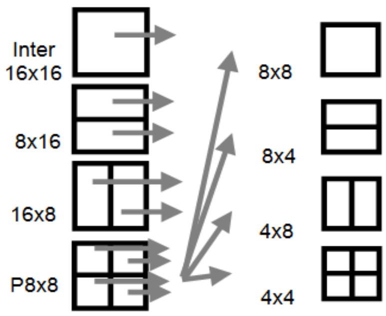


Figure 6. Different partition sizes in a MB

Secondly, H.264 usually enables the RDO method and adaptive transformation method to achieve the best combinational result of display quality and bitrate. Instead of predictive block, RDO method uses the reconstructive blocks to be the targets for choosing better coding quality and bitrate. In this way, the best combinational result of reconstructive block would be selected from all possibilities of ME and transform. Therefore, transformation block-sizes and ME's block partitions are decided at the same time.

For focusing on ATVB effect, the encoder configuration that disables P-Slice SKIP mode, Intra mode for inter slice, IPCM macroblock mode, and B coded frame. The follows is the figure of only enabling inter prediction and P frame that are the comparison computational consumption (shown as the Fig. 7), display quality (shown as the Fig. 8), and bitrate (shown as the Fig. 9).

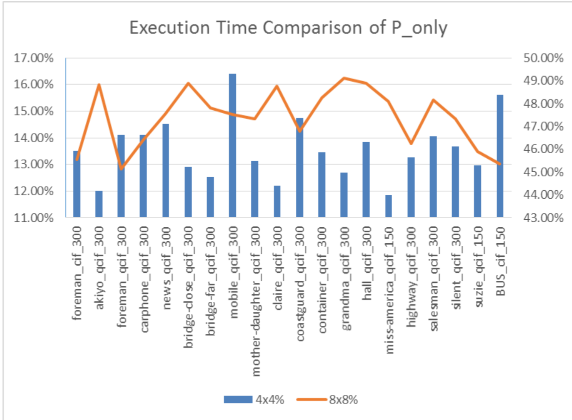


Figure 7. The execution time comparison between 4x4 or 8x8 only transform with adaptive transform, that based on inter encoding only configuration.

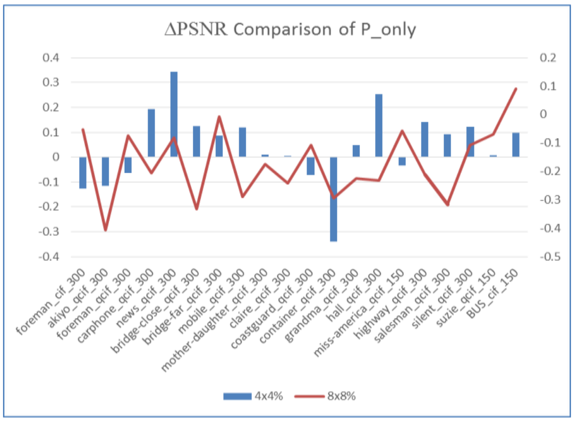


Figure 8. The difference of PSNR comparison of 4x4 and 8x8 transform by adaptive transform coding, that based on inter encoding only configuration.

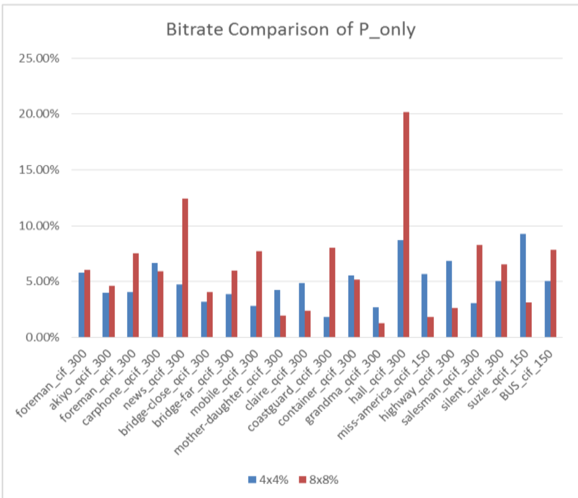


Figure 9. The bitrate comparison of 4x4 and 8x8 transform by adaptive transform coding, that based on inter encoding only configuration.

 

Fig. 10 shows the percentage of time consumption used for ME when only intra prediction and P frame are used to the encoding.

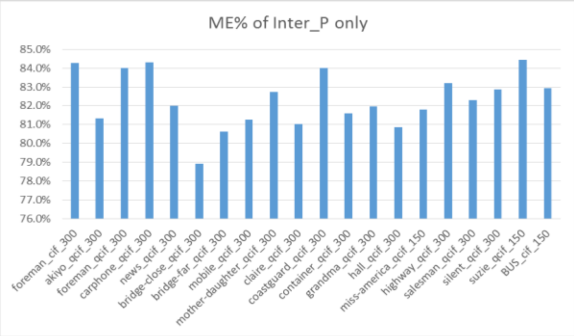


Figure 10. The execution percentates of ME with Inter_P (inter prediction) and ATVB.

The percentage of time consumption is soared, because there only adopt the inter prediction for resulting the predictive MB. The total encoding times were decreased and motion prediction times were almost same as the time with default encoding configuration. The comparison of total encoding time and ME time between the configuration of P_only and default is shown as Fig. 11.

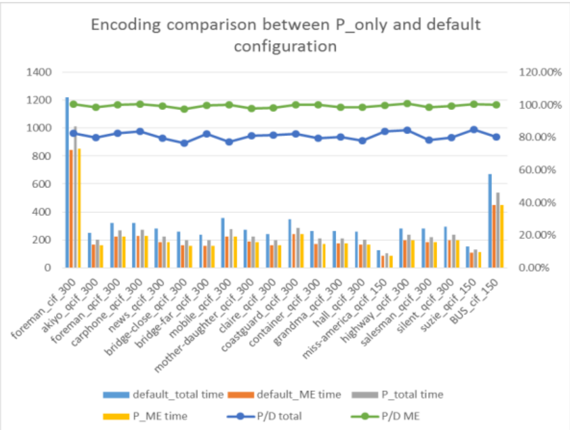


Figure 11. The comparison of P_only and default configuration based on their encoding time.

From the above figures, Fig. 9 proves ATVB can contribute a better bitrate against 4x4 or 8x8 only transform.

Thirdly, ME is the most computational module (shown as the Fig. 1 and Fig. 10 in the H.264 encoder. So, the ATVB's bitrate contribution could cause higher computation than the encoding with 4x4 or 8x8 only transform.

Fourthly, we observed that most of transformed coefficients were quantized as zero-valued samples (when QP is equal to 28). It means that a very well-predicted block was generated by ME phase.

Finally, in JM software reference, the block evaluation starts from biggest partition (of 16x16 block-size) to smallest partition (of 4x4 block-size). Another, if ATVB is enabled, motion block and transform block are decided at the same time.

Before this paper, the majority of paper [2]-[14] directly decrease the computation stand on the ME module. In lieu of this position, the paper provides a new viewpoint to offer another chance to improve current video codec.

## IV. Motivation

The paper's motivation was inspired by ATVB needs one more time of ME to achieve the best combinational result of bitrate and display quality. However, the ME is the most computational module of H.264 encoder. Hence, the speculative approach should be invented for the computational reduction. From the previous section of observation, gathering these information hopes to construct a splendid predictive method to achieve the computational reduction with ignorable display distortion.

## V. Idea and Experiment

Therefore, the paper's idea is that using the first round results of ATVB predicts if the second round evaluation needs to be proceeded. As the first point mentioned in observation, 8x8 block transform is impossible used as the corresponding ME partition is smaller than 8x8 block size. So, 8x8 block transform theoretically consumed less computation than 4x4 block transform.

According to the first and second point of observation section, the coding decisions of ATVB and ME are closely related. Adding the third observation, the benefit of display quality and bitrate brought from RDO and ATVB is very expensive, because it increases one more time computation of ME and Transform because of the coding evaluation by reconstructive block. Next, from forth point of observation, it reveals that the 8x8 block-size transform is more appropriate than 4x4 block-size transform based on the bitrate consideration when the coefficients of 8x8 or 4x4 block size are zero-valued. At the last observation, if the evaluation result of 16x16 block-size ME in the first round (with 8x8 block transform) is better than the evaluation result of 16x8 and 8x16 block-size ME, then the effect implies that the 16x16 block-size ME with 8x8 block-size transform may possible be the best coding decision.

This idea can reduce most unnecessary computation of ME and ATVB when the evaluation results of ME are ordered in ascending sequence (in which the evaluation result is ordered from16x16 MB to 4x4 sub-MB). Because ME has finished the evaluation before ATVB so that ME's evaluation result is the excellent predictive information for ATVB. When 8x8 block-size transform have been speculated to be used for encoding block, in this case, it implies the ME's predictive block whose size is smaller than 8x8 needn't to be further evaluated. In this case, the succeeding ME and Transform can be skipped for power consumption saving. In the other hand, the computation doesn't increase as ME's evaluative results are shown as descending order. There just doesn't bypass any original evaluation for ME with 4x4 transform.

The paper uses the evaluative result of ME's block partition to predict the transform's block size. Hence, the first experiment will only focus on the relationship between ME and ATVB. The JM configurations are set as (1) intra prediction disabled; (2) SKIP prediction mode disabled; (3) IPCM coding disabled; QP parameter is set as 28; (5) high complexity ME selected (that means RDO is enabled).


Figure 12. The execution time of encoder with 4x4, 8x8 ATVB, and proposal PT4S, respectively.

From Fig. 12 it shows that the execution time of proposal PT4S just higher than the one of 8x8 transform. Fig. 13 shows the reductions of execution time of 4x4, 8x8, and PT4S based on AVTB. It shows that 8x8 transform can reduce the largest amount of execution time, because the ME partitions are bigger or equal to 8x8 block that can reduce the times of ME.

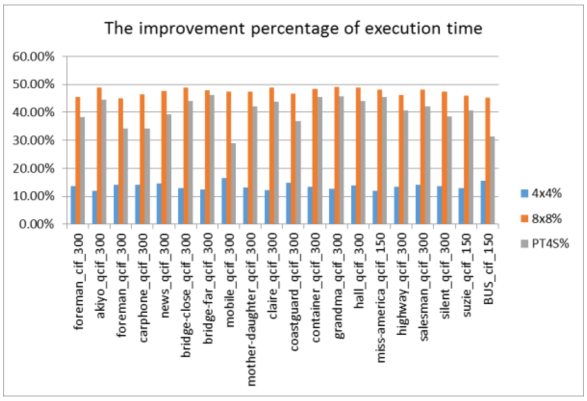


Figure 13. The reductions of execution time of 4x4, 8x8, and PT4S based on AVTB.

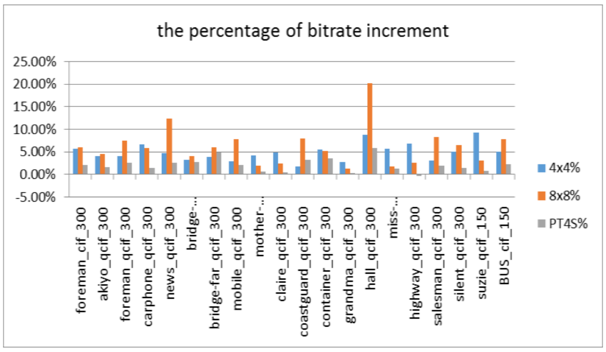


Figure 14. The percentage of bitrate increment of 4x4, 8x8, and proposed PT4S.

On the consideration of bitrate, Fig. 14 shows that PT4S encoding increases the smallest bitrate than 8x8 transform encoding. Among the most cases, PT4S encoding is better than 4x4 transform encoding. On the highway case, PT4S encoding is even better than ATVB encoding. Because the encoding results of current encoding frame are relative to previous decoded picture existed in DPB.

On the point of display quality, Fig. 15 shows the differences of PSNR encoded by 4x4 transform, 8x8 transform, and proposed PT4S that compared with ATVB encoding. Most of file encoded by proposed PT4S are resulted better quality bitstream than those encoded by 8x8 transform. And the losses of PSNR are below 0.4db between 4x4 transform and PT4S.

From the above experimental result, it is obvious to persuade that the proposal PT4S is an excellent choice for the purpose of computation reduction. The evidences as follows: a) On the computation point, PT4S skips most of unnecessarily extra execution (about 12%-25%) for ME with 4x4 transform, so that its computational complexity can be simplified as the encoder with 8x8 transform. On the bitrate point, PT4S only increases most 5% of bitrate and get the better bitrate than those encoded by 4x4 or 8x8 transform. On the PSNR point, PT4S provide proximity of display quality (that only about 0.25db losing).

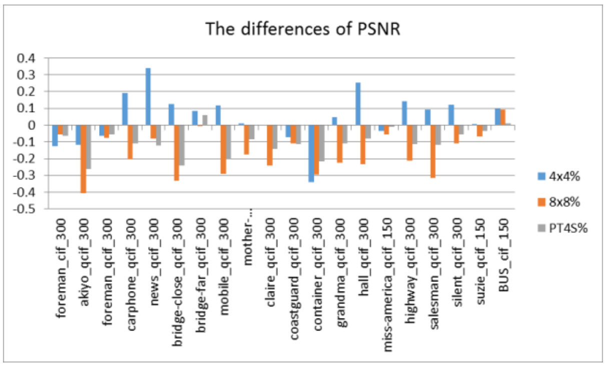


Figure 15. The differences of PSNR by the transform encoding of 4x4 block, 8x8 block, and proposed PT4S.

## VI. Conclusions and Futrure Work

From the above experimental results, they prove that PT4S is an intelligent algorithm to reduce 30%-45% computational complexity of ATVB without considerable display distortion (about maximal 0.25db loss) and bitrate increment (about maximal 2.5% bitrate increment). It is a light-weight modification with wonderful improvement and is also acceptable and easy to be included into the any version of video encoder.

The future work will add the experimental results of other higher resolution, such as 525p, 720p, and 1080p, to prove the applicable to the file of different resolutions.

## References

[1] Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264/ ISO/ IEC14496-10 AVC), Mar. 2003.

[2] R. Li, B. Zeng, and M. L. Liou, "A new three-step search algorithm for block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, no. 4, pp. 438-442, Aug. 1994.

[3] L.-M. Po and W.-C. Ma, "A novel four-step search algorithm for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 3, pp. 313-317, Jun. 1996.

[4] L.-K. Liu and B. Feig, "A block-based gradient descent search algorithm for block motion estimation in video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 4, pp. 419-422, Aug. 1996.

[5] J. Y. Tham, S. Ranganath, M. Ranganath, and A. A. Kassim, "A novel unrestricted center-biased diamond search algorithm for block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 4, pp. 369-377, Aug. 1998.

[6] A. M. Tourapis, O. C. Au, and M. L. Liou, "Predictive motion vector field adaptive search technique (PMVFAST) - Enhancing block based motion estimation," in *Proc. Visual Communications and Image Processing 2001 (VCIP-2001)*, San Jose, CA, Jan. 2001.

[7] K. Cheung and L. M. Po, "A hierarchical block motion estimation algorithm using partial distortion measure," in *Proc. International Conference on Image Processing*, Apr. 1997, pp. 606-609.

[8] J. B. Xu, L. M. Po, and C. K. Cheung, "Adaptive motion tracking block matching algorithms for video coding," *IEEE Trans. Circuits and Syst. Video Technol.*, vol. 97, pp. 1025-1029, Oct. 1999.

[9] C. Zhu, X. Lin, and L.-P. Chau, "Hexagon-Based search pattern for fast block motion estimation," *IEEE Trans. Circuits and Syst. Video Techonol.*, vol. 12, no. 5, May 2002

[10] P. Yin, H. Y. Cheong, A. M. Tourapis, and J. Boyce, "Fast mode decision and motion estimation for JVT/H.264," in *Proc. IEEE ICIP*, Sep. 2003, pp. 853-856.

[11] Y. Cheng, S. Xie, J. Guo, Z. Wang, and M. Xiao, "A fast inter mode selection algorithm for H.264," in *Proc. 1st Int. Symp. on Pervasive Computing and Applications*, Aug. 2006, pp. 821-824.

[12] Y. W. Huang, *et al.*, "Analysis and reduction of reference frames for motion estimation in MPEG-4 AVC/JVT/H.264," in *Proc. IEEE Int. Conf. on Multimedia and Expo (ICME '03)*, Jul. 2003, pp. 809-812.

[13] Y. M. Lee, Y. F. Wang, J. R. Wang, and Y. Lin, "An adaptive and efficient selective multiple reference frames motion estimation for H.264 video coding," in *Advances in Image and Video Technology*, T. Wada, F. Huang, and S. Lin, Eds., Heidelberg: Springer, 2009, vol. 5414, pp. 509-518.

[14] B. Zhan, B. Hou, and R. Sotudeh, "An efficient fast mode decision algorithm for H.264/AVC intra/inter predictions," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP '08)*, Las Vegas, NV, Mar. 2008, pp. 1057-1060.

[15] H.264/AVC JM reference software. [Online]. Available: http://iphome.hhi.de/suehring/tml/


**Kuo-Hsiang Chou** received the B.E. degree in Computer Science and Information Engineering from the Feng-Chia University, Tainan, Taiwan, Republic of China in 2002, and MS degrees in Computer Science and Information Engineering from the National Chiao Tung University, Taiwan, in 2007. He is currently pursuing for his PhD degree in Computer Science at the National Chiao Tung University, Taiwan. His research interests include video codec, computer architectures, parallel processing, and embedded system design.

**Chung-Ping Chung** received the B.E. degree from the National Cheng-Kung University, Tainan, Taiwan, Republic of China in 1976, and the M.E. and Ph.D. degrees from the Texas A&M University, Texas, USA in 1981 and 1986, respectively, all in Electrical Engineering. He was a lecturer in electrical engineering at the Texas A&M University while working towards the Ph.D. degree. Since 1986 he has been with the Department of Computer Science at the National Chiao Tung University (NCTU), Hsinchu, Taiwan, ROC, where he is a professor and the associate dean of the College of Computer Science. From 1998, he was on leave from the university and joined the Computer and Communications Laboratories, Industrial Technology Research Institute, ROC as the Director of the Advanced Technology Center, and then the Consultant of the General Director's Office, until 2002. From 2007 to 2011, he was the director of Institute of Biomedical Engineering at NCTU. He also served as the Editor-in-Chief in the Information Engineering Section of the Journal of the Chinese Institute of Engineers (EI, SCI), ROC, in 2000 to 2005, and the editor of the Journal of Information Science and Engineering (SCI) in 2006 to 2012. He has led the Computer Systems Laboratory in CS department, NCTU since 1992, served as the consultant and reviewer for numerous information and IC companies and government organizations, published over two hundred refereed technical papers, and obtained over twenty patents. His research interests include computer architecture, parallel processing, embedded system and SoC design, and parallelizing compiler.