Compensated Sum of Absolute Difference for Fast H.264 Inter Mode Selection

Lai-Man Po, Yusuf Md. Salah Uddin, Kai Guo, and Liping Wang
Department of Electronic Engineering, City University of Hong Kong,
Kowloon, Hong Kong SAR, China
E-mail: eelmpo@cityu.edu.hk

***Abstract*—In this paper, a new compensated sum of absolute difference (CSAD) for fast H.264 inter mode selection algorithm is proposed. The main idea is to determine the best inter mode based on CSAD cost instead of the rate-distortion (RD) cost. This approach can avoid most of the computationally intensive processes in the H.264 mode decision. The CSAD could solve the problem of SAD and SATD costs used in mode decision which normally bias to the smaller block size modes. It is because they are normally achieving higher prediction accuracy but consume more bit rate. Experimental results show that the proposed CSAD-based mode decision algorithm can reduce 60% to 68% of the H.264 total encoding time with negligible degradation in the RD performance.**

***Index Terms*—** Video Coding, H.264/AVC, Fast Mode Decision**.**

I. Introduction

The newest video coding standard is known as H.264 [1], which greatly outperforms the previous MPEG-1/2/4 and H.261/263 standards in terms of both picture quality and coding efficiency. To achieve this superior coding performance, H.264 adopts many advanced techniques, such as directional spatial prediction for intra frame coding, variable and hierarchical block transform, arithmetic entropy coding, multiple reference frame motion compensation, etc. It also uses seven different block sizes (16x16, 16x8, 8x16, 8x8, 8x4, 4x8 and 4x4) for motion compensation in inter mode coding, and two different block sizes with various spatial directional prediction modes in intra modes. The main purpose for employing variable block size coding is that large block modes (such as 16x16, 16x8 and 8x16) can be used for stationary image block prediction with high coding efficiency while small block modes can be used for high or complex image blocks with better prediction accuracy.

In the last few years, a number of fast mode decision algorithms [3]-[8] were developed to reduce the encoding complexity of H.264. Some of the aforementioned fast mode decision algorithms try to classify the MB into large or small partitions and skip checking some unnecessary modes. However, they still need to compute the RD costs of some possible modes for the ultimate mode decision, which involve computationally intensive processes of image transformation, quantization, entropy coding, and reconstruction. In H.264 video coding standard, SAD-based and SATD-based cost functions [10] have been developed as fast mode decision techniques for avoiding these computationally intensive processes. However, the major drawback is that the RD performance of the encoded video is quite degraded, which affects their practical implementation. In this paper, a new compensated sum of absolute value (CSAD) based fast mode decision algorithm is proposed, which can avoid RD cost computation while maintaining high RD performance.

The rest of the paper is organized with section 2 introducing the RD costs for H.264. The proposed CSAD and its fast mode decision algorithm are presented in sections 3 and 4, respectively. The parameters selection of the proposed algorithm is described in section 5. Simulation results are presented in section 6 and the conclusion is given in section 7.

II. Rate-Distortion Costs for H.264

In H.264 encoding process, the best MB coding mode is selected by computing the RD cost of all possible modes. The best mode is the one with minimum RD cost and this cost is defined as

$$J_{RD}(\mathbf{S},\mathbf{C}) = \mathrm{SSD}(\mathbf{S},\mathbf{C},) + \lambda \cdot R \qquad (1)$$

where, λ is the Lagrangian multiplier. The R is the number of bits for encoding the header information, motion vectors and quantized residual block, respectively. In equation (1), SSD(**S**, **C**) is the sum of the squared difference (SSD) between the original blocks **S** and the reconstructed block **C**, and it can be expressed as:

$$\mathrm{SSD}(\mathbf{S},\mathbf{C}) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}(s_{ij} - c_{ij})^2 = \|\mathbf{S}-\mathbf{C}\|_F^2 \qquad (2)$$

where, s_{ij} and c_{ij} are the (i, j)th elements of the current original block **S** and the reconstructed block **C**, respectively. Moreover, N is the image block size (N =4 in H.264 standard) and $\| \, \|_F$ is *Frobenius* norm. It is also found that the computation of spatial-domain SSD is very time-consuming [9]. To accelerate the coding process, the JVT reference software provides a fast SAD-based cost function:

$$J_{\text{SAD}} = \begin{cases} \text{SAD}(\mathbf{S},\mathbf{P}) + \lambda_1 \cdot 4K & \text{if intra } 4\times 4 \text{ mode} \\ \text{SAD}(\mathbf{S},\mathbf{P}) & \text{otherwise} \end{cases} \quad (3)$$

where, SAD(**S**,**P**) is sum of absolute difference between the original block **S** and the predicted block **P**. The λ_I is also approximate exponential function of the QP (quantization parameter), which is almost the square of λ, and the K equal to 0 for the probable mode and 1 for the other modes. This SAD based cost function could save a lot of computations as the distortion part is based on the differences between the original block and the predicted block. However, the expense of the computation reduction usually comes with quite significant degradation of coding efficiency. Thus, SAD is not an appropriate criterion to determine the best mode. Actually, the SAD values of different inter modes usually have the following relation: $\text{SAD}_{\text{inter16x16}} \geq \text{SAD}_{\text{inter16x8}}$ ($\text{SAD}_{\text{inter8x16}}$)$\geq \text{SAD}_{\text{inter8x8}}$. The reason behind is that the small partition motion estimation can always provide better prediction accuracy than the large mode motion estimation. According to this relationship, the J_{SAD} cost function is inclined to choose smaller block mode as the best mode since their SAD values are smaller. Thus, it can explain why J_{SAD} cost function is not appropriate for the best mode selection.

III. RELATIVE SUM OF ABSOLUTE DIFFERENCE

The reason why H.264 employs variable block-size modes is that sometimes the current MB cannot be well predicted by large partitions, which means the difference between the current MB and predicted MB is quite large. Thus, the MB should be divided into smaller partitions for better motion compensation. On the other hand, if the current block can be well predicted by the large block mode, it is unnecessary to encode the block with smaller partitions which require more encoding complexity and more bits to represent the motion vectors and side information. Thus, we need to use an appropriate criterion to tell whether the current MB achieves adequate prediction accuracy or not. SAD value presents the difference between the current MB and predicted MB, so it can reflect the prediction accuracy of different modes. As mentioned in previous section, however, J_{SAD} cost function is not appropriate for mode decision since smaller modes usually lead to smaller SAD values so that smaller block mode is inclined to be considered as the best mode. Thus, we should admit the intrinsic SAD differences between small partitions and large partitions. In addition, we should realize that the SAD values are also significant in mode decision. A small SAD difference among different modes indicates that large partition may well predict the current MB without splitting into small partitions; while a large SAD difference means the prediction accuracy of small modes outperforms that of the large mode, so that the small mode is likely to be the best mode. Since the SAD difference covers a wide range and fluctuates greatly, it is difficult to claim whether the SAD difference is small or large. Thus, we can define a new parameter, relative SAD (RSAD), to measure the SAD difference between two modes with nomination with the first SAD. The RSAD between two modes, mode 1 (representing large partition) and mode 2 (representing small partition), is defined as below:

$$\text{RSAD}_{1->2} = \frac{\text{SAD}_1 - \text{SAD}_2}{\text{SAD}_1} \quad (4)$$

where, $RSAD_{1->2}$ represents the relative prediction accuracy between mode 1 and mode 2. RSAD represents the fractional SAD difference between two modes. According to the definition of RSAD, a small RSAD of 0.1 indicates only 10% of SAD_1 difference between two modes in terms of SADs, which means that their prediction accuracy may be very similar. In this case, mode 1 is likely to be a better choice since it needs fewer bits to encode its motion vectors. On the other hand, a large RSAD implies that mode 2 is more likely to be the better mode since its prediction accuracy obviously outperforms the large partition mode 1. RSAD is a new measure to select the better mode instead of the conventional costs such as SAD, SATD and RD cost.

Based on the above idea, a threshold F can be predefined to tell whether the RSAD is small or large. If $RSAD_{1->2} < F$, select the large partition mode 1 as the better mode; Otherwise select the small partition mode 2 as the better mode. It is obvious that F affects the probability of mode 1 or mode 2 to be the selected. When F increases, mode 1 is more likely to be the better mode while a low F making mode 2 more likely to be the better mode.

How to select the threshold F is vital to this RSAD-based mode decision algorithm and which have to be robust for different types of block contents. In addition, F should also be adaptive to SAD of the mode 1 (SAD_1) as it indicates whether the large partition block is well predicted or not. If SAD_1 is small due to mode 1 is already well predicted then it is more likely to be the better mode as a result we can use a larger F. In contrast, if SAD_1 is large, the mode 2 with small partition should have higher chance to be the better mode and F should be smaller. To implement this idea, we can use another threshold T to distinguish whether SAD_1 is small or large. Similarly, T should also be adaptive and related to the QP factor of the H.264 codec. It is because a higher QP leads to rougher quantization; thus the error between the original block and reconstructed block will increase, which has a negative influence on the motion estimation performance. Then the overall SAD values will be higher, so the threshold F should be also increased. We will discuss how select these thresholds with different QP factors and show their robustness in section 5.

IV. CSAD Based Inter Mode Decision

Let us first introduce a RSAD-based mode selection algorithm for two modes only, which can be summarized as:

Step I: Initialize QP value, T, F_S , F_L based on the QP.

Step II: Calculate the SADs of mode 1 and mode 2: SAD_1 and SAD_2.

Step III: If $SAD_1 < T$, $F = F_S$; else $F = F_L$

Step IV: Calculate $RSAD_{1\to2} = (SAD_1 - SAD_2) / SAD_1$

Step V: If $RSAD_{1->2} < F$, choose mode 1 as the better mode; Else, choose mode 2 as the better mode.

In this algorithm, the F_S and F_L represent the RSAD thresholds for small SAD and large SAD blocks, respectively. On the other hand, the above algorithm is very easy to demonstrate the concept of the RSAD for mode decision, but it is not very suitable for practical implementation with more than two modes. The selection process will be very complex for selecting the best among many modes. This is also the case in H.264, which supports 7 inter mode of 16x16, 16x8, 8x16, P8x8 (8x8, 8x4,4x8, and 4x4). To simplify the selection process, the comparison of $RSAD_{1\to2}$ with the threshold F can be rearranged as comparison between SAD_1 and a compensated SAD_2 ($CSAD_2$) which is expressed as

$$CSAD_2 = SAD_2 + F.SAD_1 \quad (5)$$

Because, if $RSAD_{1->2}<F$, then $SAD_1<SAD_2+F\bullet SAD_1$. With this formulation, the RSAD threshold F becomes a compensation factor for the $CSAD_2$ with reference to the SAD_1. The major advantage of using this formulation is that we can select the best mode using $RSAD_{16x16->16x8}$, $RSAD_{16x16->8x16}$ and $RSAD_{16x16->P8x8}$ in H.264 by choosing the minimum from $CSAD_{16x16}$, $CSAD_{16x8}$, $CSAD_{8x16}$, and $CSAD_{P8x8}$ with use of several compensation factors for each CSAD. Thus, a CSAD-based mode decision algorithm for H.264 can be implemented as below algorithm for Inter modes 16x16, 16x8, 8x16, P8x8 selection:

Step I: Initialize T_{16x16}, F_{S16x8}, F_{S8x16}, F_{S8x8}, F_{L16x8}, F_{L8x16}, F_{L8x8} based on the QP.

Step II: Calculate the SAD costs of these four inter modes: SAD_{16x16}, SAD_{16x8}, SAD_{8x16} and SAD_{P8x8}.

Step III: If $SAD_{16x16} < T_{16x16}$, then
$CSAD_{16x16} = SAD_{16x16}$
$CSAD_{16x8} = SAD_{16x8}+F_{S16x8}\bullet SAD_{16x16}$
$CSAD_{8x16} = SAD_{8x16}+F_{S8x16}\bullet SAD_{16x16}$
$CSAD_{P8x8} = SAD_{P8x8}+F_{S8x8}\bullet SAD_{16x16}$
Else
$SAD_{16x16} = SAD_{16x16}$
$CSAD_{16x8} = SAD_{16x16}+F_{L16x8}\bullet SAD_{16x16}$
$CSAD_{8x16} = SAD_{16x8}+F_{L8x16}\bullet SAD_{16x16}$
$CSAD_{P8x8} = SAD_{P8x8}+F_{L8x8}\bullet SAD_{16x16}$

Step IV: Select the best mode as the minimum of ($CSAD_{16x16}$,$CSAD_{16x8}$, $CSAD_{8x16}$, $CSAD_{P8x8}$).

where T_{16x16} is SAD threshold for classifying small or large SAD_{16x16}. F_{S16x8}, F_{S8x16}, and F_{S16x8} are compensation factors for partition with small SAD_{16x16} and F_{L16x8}, F_{L8x16}, and F_{L16x8} are compensation factors for partitions with large SAD_{16x16}. Theoretically, we can also have a compensation factor for $CSAD_{16x16}$ but it is equal zero so we don't include in the above formulation. Based on this algorithm, we can determine the best inter mode by comparing the SAD of largest partition mode (Inter 16x16) and CSADs of the smaller partition modes. In this simplified algorithm, the CSAD has been implemented to only select the best mode from four Inter modes. Actually, in regard to the Inter P8x8 mode, it contains four sub-modes: Inter S8x8, Inter S8x4, Inter S4x8 and Inter S4x4 modes. Similarly, we can implement the proposed CSAD-based mode decision algorithm to choose the best sub-mode among all possible sub-modes with another set of thresholds. The CSAD-based sub mode decision process can be summarized as:

Step I: Initialize T_{8x8}, F_{S8x4}, F_{S4x8}, F_{S4x4}, F_{L8x4}, F_{L4x8}, F_{L4x4} based on the QP.

Step II: Calculate the SAD costs of these four inter modes: SAD_{s8x8}, SAD_{s8x4}, SAD_{s4x8} and SAD_{s4x4}.

Step III: If $SAD_{s8x8} < T_{8x8}$, then
$CSAD_{s8x8} = SAD_{s8x8}$
$CSAD_{s8x4} = SAD_{s8x4}+F_{S8x4}\bullet SAD_{s8x8}$
$CSAD_{s4x8} = SAD_{s4x8}+F_{S4x8}\bullet SAD_{s8x8}$
$CSAD_{s4x4} = SAD_{s4x4}+F_{S4x4}\bullet SAD_{s8x8}$
Else
$CSAD_{s8x8} = SAD_{s8x8}$
$CSAD_{s8x4} = SAD_{s8x4}+F_{L8x4}\bullet SAD_{s8x8}$
$CSAD_{s4x8} = SAD_{s4x8}+F_{L4x8}\bullet SAD_{s8x8}$
$CSAD_{s4x4} = SAD_{s4x4}+F_{L4x4}\bullet SAD_{s8x8}$

Step IV: Select the best mode as the minimum of ($CSAD_{s8x8}$, $CSAD_{s8x4}$, $CSAD_{s4x8}$, $CSAD_{s4x4}$)

where T_{8x8} is the SAD threshold for classifying 8x8 partitions with small or large SAD_{s8x8}. F_{S8x4}, F_{S4x8}, and F_{S4x4} are compensation factors for partition with small SAD_{S8x8} and F_{L8x4}, F_{L4x8}, and F_{L4x4} are compensation factors for partitions with large SAD_{s8x8}.

V. PARAMETERS SELECTIONS

As mentioned in section 3, the selection of the RSAD thresholds (or compensation factors) and thresholds for classifying small and large SAD of the large partition are vital for the proposed CSAD-based mode decision algorithm. In this section, we will discuss how to select those thresholds and show that they are not sensitive to different video content. Normally, $F_{S16x8} = F_{S8x16}$ and $F_{L16x8} = F_{L8x16}$ as they have same partition size. Similarly, $F_{S8x4} = F_{S4x8}$ and $F_{L8x4} = F_{L4x8}$ as they also have same partition size. They are, therefore, assumed to be equal in our simulations. In the rest of this paper, F_{S16x8}, F_{L16x8}, F_{S8x4} and F_{L8x4} also represent F_{S8x16}, F_{L8x16}, F_{S4x8} and F_{L4x8} respectively. To evaluates the best thresholds for the proposed CSAD-based mode decision algorithm, at first we do not consider the sub-mode related parameters and only consider T_{16x16}, F_{S16x8}, F_{S8x8}, F_{L16x8}, F_{L8x8}. In the other words, the best inter mode is determined by CSAD and the best sub-mode is determined by the conventional RD cost function. Here, we use the iterative searching method to obtain the optimal thresholds by the following algorithm:

Step I: Initialize the threshold $F_{S16x8}(0)$, $F_{S8x8}(0)$, $F_{L16x8}(0)$, $F_{L8x8}(0)$ and $T_{16x16}(0)$ for a specific QP value;

Step II: Adjust the thresholds F_{S16x8}, F_{S8x8}, F_{L16x8}, and F_{L8x8} in order to achieve local minimum RD cost. Denote the updated threshold as $F_{S16x8}(i)$, $F_{S8x8}(i)$, $F_{L16x8}(i)$, $F_{L8x8}(i)$;

Step III: Adjust the previous threshold T_{16x16} in order to achieve smaller RD cost. Denote the updated threshold as $T_{16x16}(i)$;

Step IV: Repeat Step 2 and Step 3 until the RD cost cannot be further reduced. Record the optimal threshold values.

Based on the extensive simulation upon various types of video contents (low motion, medium motion and high motion), we find that the selection of the thresholds F_{S16x8}, F_{S8x8}, F_{L16x8}, and F_{L8x8} is nearly not varied with the video contents as shown in Table I. In addition, the relationship between T_{16x16} and QP is also very robust, which is shown in Table III. It is because almost all thresholds obtained by iterative search are identical down to second digit. In addition, the optimal thresholds F_{S8x4}, F_{S4x4}, F_{L8x4}, F_{L4x4} and T_{8x8} for sub-mode selection can also be obtained via the iterative searching method. These results are shown in Tables II and IV. Similarly, these optimal thresholds based on experimental results are also nearly not varied with the video contents. Such phenomenon is also found in other video sequences and QP factors but due to the limited length of the paper, we cannot list them all. There are seem to be some nearly content-independent relation that can be used to generate these thresholds. At this moment, we cannot find simple mathematic expressions for these relationships. Fortunately, we can select a group of thresholds for (F_{S16x8}, F_{S8x8}, F_{L16x8}, F_{L8x8}) and (F_{S8x4}, F_{S4x4}, F_{L8x4}, F_{L4x4}) for practical implementation of the proposed CSAD-based mode decision algorithm, which are shown in Table V. In addition, a set of T_{16x16} and T_{8x8} thresholds are also selected based on the more extensive results of Tables III and IV for difference QPs which are shown in Tables VI and VII, respectively. A straightforward method to implement the CSAD-based mode decision algorithm is to use a table to store all these selected threshold values for different QPs and use table-lookup method to determine the required thresholds for a specified QP value.

TABLE I. THE OPTIMAL THRESHOLDS F_{S16x8}, F_{S8x8}, F_{L16x8}, AND F_{L8x8} FOUND BASED ON SIMULATION.

QP	16				24			
	F_{S16x8}	F_{S8x8}	F_{L16x8}	F_{L8x8}	F_{S16x8}	F_{S8x8}	F_{L16x8}	F_{L8x8}
Akiyo	0.15	0.30	0.03	0.07	0.15	0.30	0.03	0.07
Foreman	0.15	0.30	0.03	0.06	0.15	0.30	0.03	0.07
Stefan	0.15	0.30	0.03	0.07	0.15	0.30	0.03	0.07
QP	32				40			
	F_{S16x8}	F_{S8x8}	F_{L16x8}	F_{L8x8}	F_{S16x8}	F_{S8x8}	F_{L16x8}	F_{L8x8}
Akiyo	0.15	0.30	0.03	0.07	0.15	0.30	0.03	0.07
Foreman	0.15	0.31	0.03	0.07	0.15	0.30	0.03	0.07
Stefan	0.16	0.31	0.03	0.06	0.15	0.30	0.03	0.08

TABLE II. THE OPTIMAL THRESHOLDS F_{S8x4}, F_{S4x4}, F_{L8x4}, AND F_{L4x4} FOUND BASED ON SIMULATION.

QP	16				24			
	F_{S8x4}	F_{S4x4}	F_{L8x4}	F_{L4x4}	F_{S8x4}	F_{S4x4}	F_{L8x4}	F_{L4x4}
Akiyo	0.20	0.40	0.10	0.20	0.20	0.40	0.10	0.20
Foreman	0.20	0.40	0.10	0.20	0.20	0.40	0.10	0.20
Stefan	0.20	0.40	0.10	0.20	0.20	0.40	0.10	0.20
QP	32				40			
	F_{S8x4}	F_{S4x4}	F_{L8x4}	F_{L4x4}	F_{S8x4}	F_{S4x4}	F_{L8x4}	F_{L4x4}
Akiyo	0.20	0.40	0.10	0.20	0.20	0.40	0.10	0.20
Foreman	0.21	0.40	0.10	0.20	0.20	0.40	0.10	0.20
Stefan	0.19	0.40	0.10	0.20	0.20	0.40	0.10	0.20

TABLE III. THE OPTIMAL THRESHOLD T_{16x16} FOUND BASED ON SIMULATION

	QP=16	QP=24	QP=32	QP=40
Akiyo	200	400	900	1550
Foreman	200	400	850	1500
Stefan	200	400	900	1500

TABLE IV. THE OPTIMAL THRESHOLD T_{8x8} FOR P8X8 SUB-MODES FOUND BASED ON SIMULATION

	QP=16	QP=24	QP=32	QP=40
Akiyo	150	200	450	800
Foreman	150	250	450	800
Stefan	150	200	450	800

TABLE V. THE SELECTED THRESHOLDS FOR CSAD BASED MODE DECISION ALGORITHM

F_{S16x8}	F_{L16x8}	F_{S8x8}	F_{L8x8}
0.15	0.03	0.30	0.07
F_{S8x4}	F_{L8x4}	F_{S4x4}	F_{L4x4}
0.20	0.10	0.40	0.20

TABLE VI. THE RELATIONSHIP BETWEEN T_{16x16} AND QP

QP	12	14	16	18	20	22	24
T16x16	200	200	200	250	300	350	400
QP	28	30	32	34	36	38	40
T16x16	500	500	900	1100	1300	1400	1500

TABLE VII. THE RELATIONSHIP BETWEEN T_{8x8} AND QP

QP	12	14	16	18	20	22	24
T_{8x8}	150	150	150	150	150	200	200
QP	28	30	32	34	36	38	40
T_{8x8}	300	400	450	500	600	700	800

VI. EXPERIMENTAL RESULTS

The proposed CSAD-based mode decision algorithm was tested using the first 50 frames from four video sequences all in QCIF format. "Akiyo" is sequence of low spatial detail and changes in motion. "Foreman" has medium motion changes with dominant luminance changes. "Stefan" contains panning motion and has distinct fast motion changes. "Mobile" has slow panning, zooming and a complex horizontal and vertical motion. The experiment was carried out in the JVT JM9.6 encoder and the test parameters are listed as below:

- CABAC is enabled;
- GOP structure is IPPP;
- Max search range for motion estimation is 32;
- Max search range for motion estimation is 32;
- *QP* values are 20, 24, 28, 32, 36 and 40;

Compared with the original H.264 encoder in terms of the RD optimization and computation time, the proposed algorithm achieves a little RD performance degradation, as listed in Table VIII, as well as considerable computation time reduction as shown in Table IX. From the simulation results, we can find that the proposed CSAD algorithm can reduce the encoding time by around 60% to 68% in different *QP* values. Similarly, higher coding efficiency can be achieved in the large QP.

The encoding time is not the only criterion to evaluate an algorithm's efficiency. In the hardware implementation we mainly focus on the number of RD operations on different modes because the computation of RD cost is quite heavy in the coding process. From this aspect, our proposed algorithm is quite efficient since it determines the best mode only based on the RSAD cost function.

VII. CONCLUSION

In this paper, an efficient CSAD based inter mode decision algorithm is proposed for H.264. This algorithm is motivated by the fact that SAD values can reflect the prediction accuracy among different inter modes. However, directly using SAD to determine the best mode has been proved to be not a satisfied choice. Thus, we propose a new CSAD cost function as the measure to decide the best inter modes among 16x16, 16x8, 8x16 and P8x8. Therefore, the tremendous computation of the RD cost can be skipped. In addition, the parameters selections are also very robust for different types of video sequences. Experimental results indicate that the proposed algorithm reduce considerable encoding time as well as maintaining the quite similar RD performance.

ACKNOWLEDGMENT

The work described in this paper was substantially supported by a grant from City University of Hong Kong [ARG Project No.9667014].

TABLE VIII. R-D PERFORMANCE OF CSAD-BASED MODE DECISION ALGORITHM

Sequence	QP	Δ Bit Rate %	Δ PSNR –dB
Akiyo	20	-0.05	-0.05
	24	-0.31	-0.05
	32	-0.22	-0.03
	40	-0.08	0
Foreman	20	+0.17	-0.09
	24	-0.07	-0.08
	32	-0.04	-0.06
	40	+0.72	-0.02
Stefan	20	+0.82	-0.09
	24	+0.50	-0.11
	32	-0.02	-0.10
	40	+0.65	-0.03
Mobile	20	+0.82	-0.09
	24	+0.50	-0.11
	32	-0.02	-0.10
	40	+0.65	-0.03

TABLE IX. ENCODING TIME REDUCTION OF THE CSAD ALGORITHM

QP	20	24	32	36	40
Akiyo	-67.8%	-67.2%	-64.3%	-63.9%	-62.3%
ForMan	-64.7%	-64.4%	-61.6%	-61.4%	-60.5%
Stefan	-65.7%	-64.2%	-62.7%	-60.6%	-59.8%
Mobile	-67.8%	-66.4%	-65.0%	-61.3%	-60.7%

REFERENCES

[1] T. Wiegand, G. J. Sullivan, G.. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, pp. 560-576, July 2003.

[2] G. J. Sullivian and T. Wiegand, "Rate-distortion optimization for video compression", *IEEE Signal Process. Mag.*, vol. 15, pp.74-90, Nov. 1998.

[3] X. Jing and L. P. Chau, "Fast approach for H.264 inter mode decision," *Electron. Lett.,* vol. 40, no. 17, pp. 1051-1052, Aug. 2004.

[4] D. Wu, F. Pan, K. P. Lim and S. Wu, "Fast intermode decision in H.264/AVC video coding," *IEEE Trans. Circuits Syst. Video Technol.,* vol. 15, no. 7, pp.953-958, Jul. 2005.

[5] T. Y. Kuo and C. H. Chan, "Fast variable block size motion estimation for H.264 using likelihood and correlation of motion field", *IEEE Trans. Circuits Syst. Video Technol.,* vol. 16, pp.1185-1195, Oct. 2006.

[6] C. Grecos, M. Yang, "Fast Mode Prediction for the Baseline and Main Profiles in the H.264 Video Coding Standard", *IEEE Trans. Multimedia*, vol. 8, pp. 1125-1134, Dec. 2006.

[7] Z. P. Lin, H. T. Yu, F. Pan, "A scalable fast mode decision algorithm for H.264", *Proc. IEEE ISCAS 2006*, May 2006.

[8] Z. Y. Wei, K. N. Ngan, "A Fast Macroblock Mode Decision Algorithm for H.264", *Proc. IEEE APCCAS 2006*, pp. 772-775, Dec. 2006.

[9] A. Hallapuro, M. Karczewicz, "Low Complexity Transform and Quantization –Part I: Basic Implementation", *ITU-T Q.6/SG16 Document JVT-B038*, Jan. 2002.

[10] Joint Video Team (JVT) Reference Software version 6.1d