EFFECT OF THE HADAMARD TRANSFORM ON MOTION ESTIMATION OF DIFFERENT LAYERS IN VIDEO CODING

Abdelrahman Abdelazim[a], Martin Varley[a] and Djamel Ait-Boudaoud[b]*.

[a]School of Computing, Engineering and Physical Sciences,University of Central Lancashire, Preston. PR1 2HE. UK
{AAbdelazim, MRVarley}@ uclan.ac.uk
[b]Faculty of Technology, University of Portsmouth, Portsmouth. PO1 3AH. UK
djamel.ait-boudaoud@port.ac.uk

KEY WORDS: Video Coding, Motion Estimation, Hadamard Transform, H.264

ABSTRACT:

In video coding, the most commonly used Motion Estimation distortion metrics are predominantly based on the Sum of Absolute Differences (SAD) and the Sum of Absolute Transformed Differences (SATD). Consequently the Joint Model (JM) H.264/AVC Reference Software utilises them and by default, the JM software selects the SAD as the Error Metric for Full-Pixel (first layer) motion estimation and the SATD as the Error Metric for Half and Quarter-Pixel (second and third layer respectively) motion estimation. Although SATD is much slower than SAD, it more accurately predicts quality from the standpoint of both objective and subjective metrics. In this paper, our experimental results show that the current H.264/AVC Rate-Distortion Optimisation method can have a negative impact when the SATD is applied. More specifically, although the SATD results in a lower bit rate with the same Peak Signal to Noise Ratio (PSNR) when applied in the integer pixel motion estimation with the subpel search disabled, it does not result in a better Rate-Distortion (R-D) performance when applied in the integer pixel motion estimation with the subpel search enabled, when compared to applying the SAD in the integer pixel motion estimation with the subpel search enabled.

1. INTRODUCTION

Today's hybrid video coding techniques apply motion-compensated prediction in combination with transform coding of the prediction error. This is done to reduce the bit rate of video signals. In recent video coding standards such as H.264/AVC (Wiegand, et al., 2003) there are seven different block sizes that can be used for motion-compensated prediction. Furthermore, to enhance the coding efficiency, the standard allows quarter-sample prediction signal accuracy.

Previously, the motion-compensated prediction result that provides the minimal distortion was widely accepted as the prediction signal. However, in recent years, it has been realised that such a selection is not always the most efficient, since the minimal distortion may result in a high bit rate, thereby degrading the overall coding performance. To solve this problem, the Rate-Distortion Optimisation (RDO) concept has been introduced. RDO techniques minimise the distortion under a constraint on the rate. A classical solution to the RDO problem is the Lagrangian optimisation which is used in the H264/AVC standard. The basic idea of this technique is to convert the RDO problem from a constrained problem to an unconstrained problem.

The Lagrangian cost function is divided into two parts; Distortion and Rate. The Distortion measurement quantifies the quality of the reconstructed pictures while the Rate quantifies the bits needed to code the macroblock. The Lagrange multiplier is usually calculated in a heuristic way or in an analytical way based on Rate-Distortion (R-D) models (Wiegand & Girod 2001) & (Li, et al., 2009).

The JM software allows the user to select the motion estimation distortion metric between the Sum of Absolute Differences (SAD), the Sum of the Squared Differences (SSD) and the Sum of Absolute Transformed Differences (SATD), the latter uses the Hadamard Transform. This has been employed to improve the rate-distortion performance and to facilitate the standard to gain much support in a variety of application areas. In this paper, the implications of the SATD based Motion Estimation on different layers are discussed. Moreover, a comparison between the SAD and SATD effect on the coefficients bits and motion vector bits on different layers is presented and analysed. In addition, future work to improve the R-D performance is proposed.

The paper is organised as follows. Section 2 gives a brief overview of the motion estimation proposed in the H.264/AVC. Section 3 describes the implications of the SATD based motion estimation on different layers, details and discussion of a comprehensive list of comparative experimental results. Section 4 concludes the paper.

2. MOTION ESTIMATION IN H.264/AVC

In the first stage of ME, an integer-pixel-motion-search is performed for each square block of the slice to be encoded in order to find one (or more) displacement vector(s) within a search range. The best match is the position that minimises the Lagrangian cost function J_{motion}:

$$J_{motion} = D_{motion} + \lambda_{motion} R_{motion} \qquad (1)$$

where λ_{motion} is the Lagrangian multiplier, D_{motion} is an error measure between the candidate macroblock taken from the reference frame(s) and the current macroblock and R_{motion} stands for the number of bits required to encode the difference between the motion vector(s) and its prediction from the neighbouring macroblocks (differential coding). A similar function to equation (1) is used to decide the optimal block size for motion estimation.

The most common error measures are the Sum of Absolute Difference (SAD) and the Sum of Absolute Transformed Differences (SATD). In particular, for any given block of pixels, the SAD between the current macroblock and the reference candidate macroblock is computed using the following equation:

$$SAD = \sum_{ij} | C_{ij\,-} R_{ij} | \qquad (2)$$

where C_{ij} is a pixel of the current macroblock and R_{ij} is a pixel of the reference candidate macroblock.

After the integer-pixel-motion-search finds the best match, the values at half-pixel positions around the best match are interpolated by applying a one-dimensional 6-tap FIR filter horizontally and vertically. Then the values of the quarter-pixel positions are generated by averaging pixels at integer and half-pixel positions. Figure 1 illustrates the interpolated fractional pixel positions. Upper-case letters indicate pixels on the full-pixel grid, while numeric pixels indicate pixels at half-pixel positions and lower case pixels indicate pixels in between at quarter-pixel positions [1] and [6].

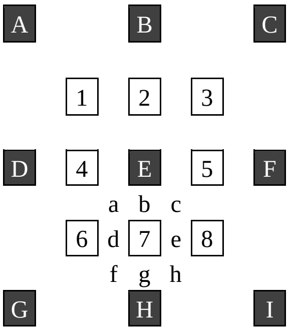

Figure 1 – *Fractional pixel search positions.*

For example, in the figure above if the integer best match is position E, the half-pixel positions 1, 2, 3, 4, 5, 6, 7, 8 are searched using equation (1). Suppose position 7 is the best match of the half pixel search. Then the quarter-pixel positions a, b, c, d, e, f, g, h are searched using again equation (1).

The Lagrangian cost can also be minimised in the frequency domain, in a very similar manner to the pixel domain. As mentioned above, SATD can be used in equation (1) instead of SAD. Central to the calculation of SATD is the 4x4 Hadamard transform which is an approximation to the 4x4 DCT transform. The transform matrix used is shown in Figure 2 below (not normalised):

$$\mathbf{H} = \begin{array}{|c|c|c|c|} \hline 1 & 1 & 1 & 1 \\ \hline 1 & 1 & -1 & -1 \\ \hline 1 & -1 & -1 & 1 \\ \hline 1 & -1 & 1 & -1 \\ \hline \end{array}$$

Figure 2 – *Hadamard Transform Matrix.*

Since **H** is a symmetric matrix, it is equal to its own transpose. By using this matrix, the (SATD) is computed using equation (3) below:

$$SATD = (\sum_{i,j} | H * (C_{ij} - R_{ij}) * H |) / 2 \qquad (3)$$

where C_{ij} and R_{ij} are the same as in equation (2) and H is the matrix in figure 2. The reader should note that the application of the Hadamard transform is optional in any resolution and can be enabled/disabled in the configuration files of the standard.

3. THE IMPLICATIONS OF THE SATD-BASED MOTION ESTIMATION ON DIFFERENT LAYERS

3.1 Use of SATD in video coding

In hybrid video coding approach following the Motion Estimation that exploits temporal statistical dependencies as described in section 2, a transform coding of the prediction residual is performed to exploit spatial statistical dependencies.

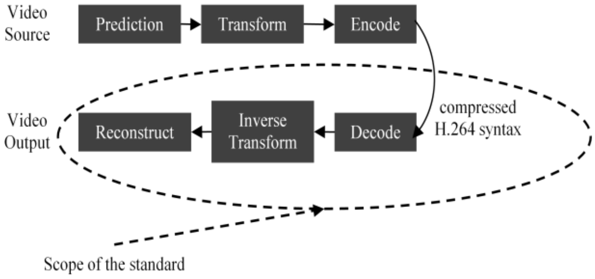

Figure 3 – *Scope of video coding standardisation*

For transform coding purposes, each colour component of the prediction residual signal is subdivided into smaller 4x4 blocks. Each block is transformed using an integer transform, and the transform coefficients are quantized and encoded using entropy coding methods.

In H.264/AVC, the transformation is applied to 4x4 blocks, and instead of a 4x4 Discrete Cosine Transform (DCT), a separable integer transform with similar properties as a 4x4 DCT is used. The transform matrix is shown in figure 4.

$$\mathbf{H} = \begin{array}{|c|c|c|c|}\hline 1 & 1 & 1 & 1 \\ \hline 2 & 1 & -1 & -2 \\ \hline 1 & -1 & -1 & 1 \\ \hline 1 & -2 & 2 & -1 \\ \hline \end{array}$$

Figure 4 – I*nteger Transform Matrix.*

The Hadamard transform is the simplest orthogonal transform and eliminates spatial redundancies of image therefore it is usually considered as some kind of coarse approximation of DCT. This can be clearly realised when figure 2 and figure 4 are compared. As a result, ME combined with Hadamard transform is expected to find optimal difference blocks with lower redundancies, which are more suitable for subsequent DCT coding.

3.2 Effect of the Hadamard transform on motion estimation of different layers

Although the above holds true when the SATD is simply compared to other error measures metrics, in the H.264 the implementation is far more complicated; as there are two main factors that affect the overall performance. The first one is the Lagrangian cost function and its associate Lagrange multiplier and the second one is the interpolation filters that provide the half-pixel and quarter pixel search position (Wedi & Musmann 2003).

In this section, to illustrate the effects of these factors two set of experiments have been carried out. Firstly, to demonstrate the effect of the Lagrangian cost function on the SATD, we disabled the subpixel Motion search and compared the SATD performance to the SAD performance in terms of the Bjontegaard Delta Bit Rate (BDBR) percentage differences and the Bjontegaard Delta PSNR (BDPSNR) differences (in dB) (Bjontegaard, 2001), the total encoding time differences and the difference in the Distortion Weight (DW) in the Lagrangian cost function, the latter reflects the impact on the number of the required bits to encode the residual coefficients and the motion information. The result is shown in table 1. Secondly, to demonstrate the interpolation effect on the SATD, we enabled the sub-pixel motion search and performed the same comparison. The result is shown in table 2.

In this experiment six kinds of video sequences with different motion characteristics were used. The “Akiyo” sequence shows slow motion and fixed background. “Foreman” is a sequence with medium changes in motion and contains dominant luminance changes. “Tempete” is a sequence of spatial detail, fast random motion and camera zoom. “Silence” is a sequence of low spatial detail and medium changes in the motion of the arms and head of the person in the sequence. “Stefan” contains panning motion and has distinct fast changes in motion. “Mobile” contains slow panning, zooming, a complex combination of horizontal and vertical motion and high spatial colour detail.

The chosen search range was 32 pixels for the full motion estimations in the H.264 ‘baseline’ profile. The configuration file for the encoder had the following settings: Level 40, RD optimisation ON, IPPP structure, CABAC coding, and the number of reference slices was 5.

In these experiments, the source code for the H.264 Reference Software Version JM14.2 (Sühring, 2008) was used. Two sizes QCIF (176×144) & CIF (352×288) were used in an Intel Core 2 CPU 6420 @ 2.13 GHz with 3.0 GB RAM.

For clarification purposes, the measures used in the tables are briefly explained here. The minus signs denote PSNR degradation and bitrate savings respectively. Encoding Time increase is computed as follows:

$$\Delta Time = \frac{Time_{SATD} - Time_{SAD}}{Time_{SAD}} \times 100\% \qquad (4)$$

The difference in the Distortion Weight (DW) in the Lagrangian cost function is calculated as follows:

From equation (1) DW_{SAD} and DW_{SATD} are calculated using equation (5) & (6) respectively.

$$DW_{SAD} = \frac{\sum\limits_{The\_whole\_sequence} SAD}{\sum\limits_{The\_whole\_sequence} SAD + \sum\limits_{The\_whole\_sequence} \lambda_{motion} R_{motion}} \times 100\% \qquad (5)$$

$$DW_{STAD} = \frac{\sum\limits_{The\_whole\_sequence} STAD}{\sum\limits_{The\_whole\_sequence} STAD + \sum\limits_{The\_whole\_sequence} \lambda_{motion} R_{motion}} \times 100\% \qquad (6)$$

Then the difference is calculated using equation (7)

$$\Delta DW = \frac{DW_{SATD} - DW_{SAD}}{DW_{SAD}} \times 100\% \qquad (7)$$

Sequence	size	BDPSNR (db)	BDBR (%)	Time (%)	DW (%)
Akiyo	QCIF	+0.1	-2.1	1053	115.9
	CIF	+0.07	-1.85	1241	144.2
Foreman	QCIF	+0.14	-3.2	710.2	60.2
	CIF	+0.16	-4.16	865.2	63
Mobile	QCIF	+0.12	-1.3	474.9	19.8
	CIF	+0.1	-1.45	623.3	29.7
Stefan	QCIF	+0.08	-1.1	524.5	23.4
	CIF	+0.07	-1.12	665.8	30.6
Silent	QCIF	+0.05	-1.1	900.9	67.1
	CIF	-0.06	-1.6	1055	75.6
Tempete	QCIF	+0.1	-1.5	543.7	28
	CIF	+0.1	-1.72	714.9	75.6
Average		+0.1	-1.85	781	61.1

Table 1– *Comparison on (BDPSNR), (BDBR), encoding time and the difference in the Distortion weight in the Lagrangian cost function between the SAD and SATD when subpixel Motion search is disabled*

Sequence	size	BDPSNR (db)	BDBR (%)	Time (%)	DW (%)
Akiyo	QCIF	-0.01	+0.07	999	118
	CIF	+0.01	+0.1	1149	141
Foreman	QCIF	-0.05	+1.05	673.9	56.72
	CIF	-0.01	+0.12	835.4	60.53
Mobile	QCIF	-0.017	+0.2	482.2	20.3
	CIF	0.1	+1.1	661	28.7
Stefan	QCIF	+0.01	+0.01	539.8	23.3
	CIF	0	+0.01	671.3	29.8
Silent	QCIF	-0.01	+0.13	865	66.3
	CIF	-0.1	+0.31	1023	74.3
Tempete	QCIF	+0.01	-0.02	541.3	29.1
	CIF	+0.1	1.65	681	41.1
Average		+0.01	+0.39	760.1	57.4

Table 2– *Comparison on (BDPSNR), (BDBR), encoding time and the difference in the Distortion weight in the Lagrangian cost function between the SAD and SATD when subpixel Motion search is enabled.*

Table 1 shows the bitrate percentage differences (BDBR) average is -1.85 while the Delta PSNR (BDPSNR) differences average is +0.1 dB. This indicates that although the Hadamard transform outperforms the SAD, it doesn't have a significant impact on the RD performance. The reason for that can also be seen from the table where the average DW value when SATD is used is approximately 60% greater than the average DW value when SAD is used. This reduces the contribution of the second part in equation (1) and produces higher motion vector bits.

Table 2 illustrates the negative effect of the interpolation on the Hadamard transform. From the table it can be seen that when the subpixel is enabled, although when the SATD is used the average total encoding time is increased by 760%, the RD performance is degraded. Since Hadamard transform aims to match frequencies instead of pixels to get a better performance in the transform/quantisation process by reducing the coefficients bits, our observation showed that the Hadamard transform successfully reduces the coefficients bits significantly, however, in some cases the Hadamard transform does not result in finding the true motion which makes the interpolation process ineffective and affects other areas due to the prediction.

Further investigations have been carried out to examine the SATD performance against the SAD performance in subpel search. The results of these investigations indicated that the Hadamard transform outperform the SAD significantly in the subpel search because the search positions are limited to 9 positions (In full pixel the number of positions = (2*search_range+1)*(2*search_range+1)) which limit the motion vector range and increase the significance of the first part in equation (1). Furthermore, the increase in the encoding time can be tolerated.

4. CONCLUSION AND FUTURE WORK

Since ultimately the transformed coefficients are coded, better estimation of the cost can be achieved by estimating the effect of the DCT with a 4×4 Hadamard transform. Although these advantages are well known and the Hadamard transform is implemented in various parts of the ME and MD processes of the standard, for the best of our knowledge no research has been carried out to investigate the effect of the λ selection and the interpolation on the SATD. The reason for this is the extensive computations required to execute the SATD; which involves subtraction, addition, shift and absolute operations. However, if the ME is improved to accommodate the use of the SATD in the fullpixel motion search, in addition to enhancing the effect of the DCT, significant RD enhancement can be achieved in wide range of applications. Particularly, in hardware applications when applying the same distortion metric at different resolutions is essential.

To overcome some of the limitations of using SATD in the full pixel Motion Search two methods can be introduced:

1) Store a few ME vector candidates (the number can vary, subject to experiments then applying the Sum of Absolute Transformed Difference (SATD) using Hadamard transform of those candidates. This should improve the bit rate by having positive effect on the DCT and minimises the effect of the above mentioned problem.

2) Train the λ as in (Wiegand & Girod 2001), but instead of using SAD use SATD.

Further research is necessary to enhance the RD performance when SATD is used the full pixel Motion Search.

REFERENCES

Bjontegaard, G., Apr. 2001. G. Calculation of Average PSNR Differences between RD-curves. Doc. VCEG-M33.

Li, X. Oertel, N. Hutter, A. & Kaup, A.,2009. Laplace Distribution Based Lagrangian Rate Distortion Optimization for Hybrid Video Coding, *IEEE Transactions on Circuits and Systems for Video Technology*, 19(2), pp.193-205.

Sühring, K., 2007. H.264/AVC Reference Software Version JM14.2 : Joint Video Team. Available at: http://iphome.hhi.de/suehring/tml/download/ (accessed Dec. 2009).

Wedi, T. & Musmann, H.G., 2003. Motion- and Aliasing-Compensated Prediction for Hybrid Video Coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7), pp.577-586.

Wiegand, T. Sullivan, G.J. Bjøntegaard, G. & Luthra, A., 2003. Overview of the H.264/AVC Video Coding Standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7), pp.560-576.

Wiegand, T. & Girod, B., 2001 Lagrange multiplier selection in hybrid video coder control. IEEE International Conference on Image Process. (ICIP), Thessaloniki, Greece, Vol.3. pp. 542-545.