MOTION VECTOR PREDICTION FOR IMPROVING ONE BIT TRANSFORM BASED MOTION ESTIMATION

Colin Doutre and Panos Nasiopoulos

Department of Electrical and Computer Engineering
University of British Columbia, Vancouver, Canada

ABSTRACT

One Bit Transforms (1BT) have been proposed for lowering the complexity of motion estimation (ME) in video coding. These transforms generate a one bit representation of each pixel in the video that is used in the motion search. This approach can greatly reduce the silicon area and power required for hardware based video encoding. However 1BT methods under-perform traditional Sum of Absolute Differences (SAD) based motion estimation, particularly for smaller block sizes. In this paper, it is proposed to improve 1BT based ME by predicting the motion vector for each block based on the vectors from previous blocks and modifying the cost function to favor motion vectors close to the predicted one. This takes advantage of the spatial correlation between motion vectors and produces a more uniform motion field. Simulation results show the proposed method can improve the PSNR of frames reconstructed through motion compensation by up to 1 dB and substantially improve the subjective video quality by reducing blocking artifacts.

Index Terms— motion estimation, one bit transform, video coding

1. INTRODUCTION

Motion Estimation (ME) and Motion Compensation (MC) are key techniques in digital video compression. MC exploits the temporal redundancy between frames in a video, greatly improving compression efficiency. However performing ME at the encoder is one of the most computationally expensive operations involved in digital video compression, often taking over 50% of the computations performed by the encoder [1].

The most popular ME technique used in video coding applications is block matching. Block matching involves dividing a video frame into non-overlapping blocks and for each block finding a matching block in a previously coded frame. The criteria used for determining the "best" match is usually the Sum of Absolute Differences (SAD). If the frame is divided into blocks of size *NxN* pixels and the SAD matching criteria is used, a cost for each potential displacement vector *(m,n)* is calculated as:

$$SAD(m,n) = \sum_{i=i_0}^{i_0+N} \sum_{j=j_0}^{j_0+N} \left| I_t(i,j) - I_{t-1}(i-m, j-n) \right| \quad (1)$$

where $I_t(i,j)$ is the current frame, $I_{t-1}(i,j)$ is the previous frame, and (i_0,j_0) is the location of the block for which a match is being found. A motion vector ***mv*** is found for each block by minimizing the cost function:

$$(mv_x, mv_y) = \underset{-s<m,n<s}{\arg\min} \quad SAD(m,n) \quad (2)$$

where *s* is the search range. In order to find the optimal motion vector with (1) and (2), a Full Search (FS) can be used, where the SAD is calculated for every possible displacement vector within the search range. The FS algorithm is guaranteed to find the optimal motion vector, but requires a huge number of calculations.

In order to speed up the ME process, a large number of techniques have been proposed. One category of techniques is fast search methods that evaluate the cost function for a subset of the possible search locations, such as the logarithmic search [2] and Diamond Search [3]. Another set of techniques involves using different matching criteria that can be evaluated more efficiently than the SAD.

In this paper, we consider the set of techniques that employees a different matching criteria; in particular One Bit Transform (1BT) based methods. The 1BT converts each frame in the video into a one bit per pixel representation for the purposes of performing the motion search. Each pixel is classified as either a match or non-match with a pixel in the previous frame using a simple XOR operation. This greatly reduces the number of gates required for a custom hardware ME implementation compared to using SAD, because multiple bit subtraction and absolute value operations are replaced with single bit XOR operations. This can greatly reduce the silicon area and power requirements. Employing a 1BT can also greatly reduce the memory bandwidth required for ME,

1-4244-1484-9/08/$25.00 ©2008 IEEE

since a single bit representation of the reference frames will be read from memory rather than the full (typically 8-bit) representation during the memory intensive motion search.

Different methods have been proposed for generating a one bit representation of each frame. In [4], the block mean is used as a threshold for determining the one bit value of each pixel in a block. Edge detection is used in [5], where a threshold is applied to generate a binary edge map. A particularly effective method is proposed in [6], where each frame is compared to a filtered version of the frame on a pixel-wise basis. The following multi-band pass filter is used for generating the filtered frame, $I_F(i,j)$:

$$K(i,j)=\begin{cases}1/25 & i,j\in[0,4,8,12,16]\\ 0 & \text{otherwise}\end{cases} \tag{3}$$

The one-bit representation, $B(i,j)$ is generated by:

$$B(i,j)=\begin{cases}1 & I(i,j)>I_F(i,j)\\ 0 & \text{otherwise}\end{cases} \tag{4}$$

The filtered frame serves as a pixel wise threshold for generating the one-bit frame. Recently, an improvement to the method in [6] was proposed, where a similar filter is used, but with 16 non-zero elements [7]. This allows the filter to be implemented with only integer addition and bit-shift operations, avoiding the need for computationally expensive floating point multiplications. The method in [7] also gives slightly better reconstructed image quality than the method in [6] for small block sizes.

The cost function typically used for one bit ME is the Number of Non-Matching Points (NNMP). Once a one bit representation has been obtained for the current frame, $B_t(i,j)$ and previous frame, $B_{t-1}(i,j)$ the NNMP is calculated for each displacement vector (m,n) as:

$$NNMP(m,n)=\sum_{i=i_0}^{i_0+N}\sum_{j=j_0}^{j_0+N}B_t(i,j)\oplus B_{t-1}(i-m,j-n) \tag{5}$$

As with the SAD case in equation (2), a motion vector is chosen by minimizing the NNMP.

A two bit transform (2BT) for ME is proposed in [8], where a two bit representation of each pixel is obtained by comparing the pixel to the mean and variance of the surrounding block. This allows the image to be divided into four classes rather than two, improving the ME accuracy at the expense of greater complexity.

A problem with 1BT and 2BT methods is that the range of possible NNMP values is much lower than the range of possible SAD values. For 8x8 blocks the NNMP has a range of [0, 64], while for 8-bit data the SAD has a range of [0, 16320]. Consequently, when using the NNMP metric, there are often many displacement vectors that have a cost value very close to the minimum. Small amounts of noise can cause the chosen motion vector to change. So 1BT and 2BT methods are more prone to giving "bad" motion vectors that don't correspond to the true motion of the scene.

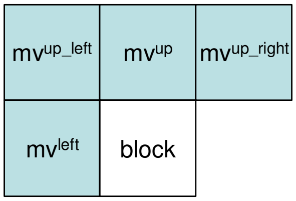

Figure 1: Motion vectors available for predicting the motion vector for the current block

In this paper we propose a method for improving 1BT and 2BT based ME by considering the neighboring motion vectors in the cost minimization process. We form a prediction for each MV based on previously determined MV's of the blocks neighbors, and penalize each potential displacement vector based on how much it differs from the predicted MV. This takes advantage of the spatial correlation between MV's and significantly decreases the number of "bad" MV's.

The rest of the paper is organized as follows. The proposed method for generating a predicted MV and modifying the cost function are described in section II. Experimental results are presented in section III and conclusions are given in section IV.

2. PROPOSED METHOD

The proposed method involves predicting the MV for each block, and penalizing MV's based on how much they differ from the predicted vector. This helps ensure that the motion field is smooth.

In most video coding standards, blocks are scanned and coded in raster order. This means that the MV's for blocks above and to the left of the current block are available for making a prediction of the current block's MV (Fig. 1). In the H.264/AVC standard, the median of the surrounding available vectors is used for motion vector prediction [9]. The problem with median prediction is that it is relatively computationally complex. We propose to use a simpler method where the mean of the left and upper blocks is used as the predicted motion vector ***mvp***:

$$(mvp_x,mvp_y)=\left(\frac{mv_x^{up}+mv_x^{left}}{2},\frac{mv_y^{up}+mv_y^{left}}{2}\right) \tag{6}$$

More previous MV's could be used in generating the predicted MV. However, experimentally we have found that there is a negligible difference in performance between the simple two element average in (6) and averages involving more terms or median based prediction.

Table I: Average PSNR (dB) of Sequences Reconstructed With Different ME Techniques using 8x8 Blocks and Full Search

Method	λ	Coastguard 299 frames	Flowergarden 114 frames	Football 124 frames	Foreman 299 frames	Mobile 139 frames	Tennis 149 frames
SAD	-	31.62	25.38	24.81	32.96	23.85	31.08
MF-1BT	-	29.75	24.48	22.87	29.98	22.74	29.01
MF-1BT MVP	0.25	30.14	24.61	23.08	30.53	22.86	29.34
	0.50	30.41	24.67	**23.16**	30.81	22.94	**29.47**
	0.75	30.48	24.68	23.14	30.88	22.94	29.46
	1.00	30.62	**24.72**	23.07	30.96	23.02	29.40
	1.50	30.63	**24.72**	22.93	**30.97**	23.04	29.29
	2.00	**30.64**	24.70	22.70	30.89	**23.06**	29.10
2BT	-	30.52	24.75	23.41	30.59	22.91	29.64
2BT MVP	0.25	30.68	24.80	**23.49**	30.92	23.01	29.74
	0.50	30.78	**24.82**	23.48	31.08	23.09	**29.75**
	0.75	30.80	**24.82**	23.43	**31.10**	23.09	29.70
	1.00	**30.86**	**24.82**	23.35	**31.10**	**23.16**	29.62
	1.50	30.84	24.80	23.12	31.02	**23.16**	29.40
	2.00	30.80	24.75	22.81	30.84	23.15	29.14

Given the predicted MV, we modify the cost function for each possible displacement vector as:

$$COST(m,n) = NNMP + \lambda\left(\left|mvp_x - m\right| + \left|mvp_y - n\right|\right) \qquad (7)$$

Our modified cost function combines the NNMP with the absolute value of the difference between the predicted motion vector and the displacement vector being evaluated. The optimal choice of the weighting factor λ will depend on several factors including the block size, the amount of motion in the video and the regularity of the motion in the video. To keep complexity low, it is desirable to have a λ that is an integer power of two so the weighting can be done with a bit-shift operation rather than a multiplication. Different values of λ are evaluated experimentally in the results section.

3. RESULTS

The proposed method of Motion Vector Prediction (MVP) and the modified cost function can be combined with virtually any ME method. Here, it is tested with the Multiplication Free One Bit Transform (MF-1BT) in [7] and the Two Bit Transform (2BT) in [8]. ME was performed on six standard test sequences with a block size of 8x8 pixels and a search range of 8 pixels. Results are presented for $\lambda \in [0.25, 0.5, 0.75, 1, 1.5, 2]$. A full search was used for all tests, where every displacement vector within the search range is tested.

Table I shows the PSNR of the test sequences when each frame is reconstructed from the previous frame using motion compensation with the motion vectors determined by the different ME methods. The highest PSNR obtained by varying λ is shown in bold for each sequence.

The results show that the PSNR of the reconstructed sequence is not very sensitive to the value of λ, particularly within the range 0.5-1.5. Using λ=1 is particularly appealing as it saves an operation, and for most cases gives a PSNR within 0.1 dB of the maximum PSNR obtained by varying λ.

When MF-1BT is used, the proposed MVP method results in gains of 0.2 to 1.0 dB in PSNR for the different sequences. When the 2BT is used, the gains obtained using the proposed method range from 0.1 to 0.5 dB. The gains in PSNR are highest for the sequences where there is the biggest performance gap between SAD and the 1BT or 2BT. For half of the test sequences, using the MF-1BT together with MVP gives better performance than the 2BT without MVP.

It is well know that PSNR does not always accurately represent the perceived image quality. Even when the gain in PSNR is modest, there can be a substantial gain in perceptual quality when using our MVP method. The MVP method helps prevent sharp changes in the motion field, which reduces blocking artifacts in the frames reconstructed with MC. An example of this is shown in Figure 2, which shows a frame of the Flowergarden sequence reconstructed with MC and the various ME techniques. For this frame, the gain in PSNR obtained through MVP is small (0.05 dB for both MF-1BT and 2BT), but visual comparison of (c)-(d) and (e)-(f) shows MVP greatly reduces blocking artifacts.

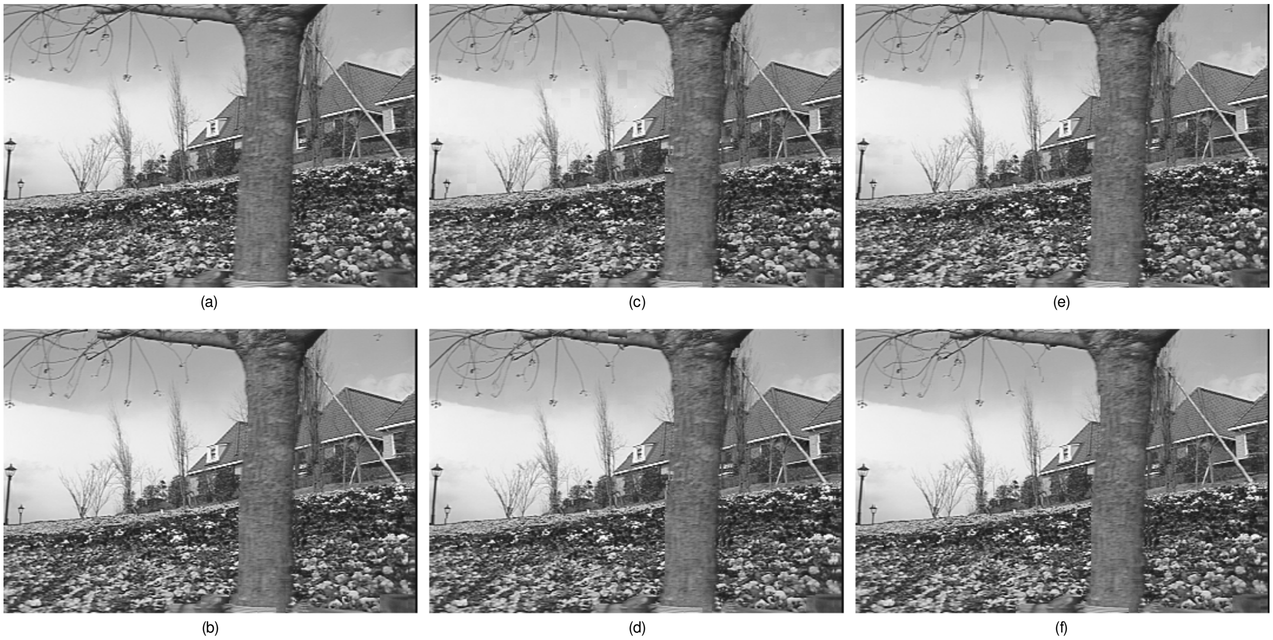

Figure 2: Frame 2 of the ‘flowergarden’ sequence (a) Original frame (b)-(f) The frame reconstructed from the previous frame using MV’s obtained by (b) SAD, PSNR = 25.35 dB (c) MF-1BT, PSNR = 24.37 dB (d) MF-1BT MVP, λ=1, PSNR = 24.42 dB (e) 2BT, PSNR = 24.58 dB (f) 2BT MVP, λ=1, PSNR= 24.63 dB

Another advantage of the MVP method is that it favors a uniform motion field, so fewer bits will be needed to code the motion vectors.

The additional complexity imposed by the proposed MVP method is as follows. For each block, 2 addition and 2 bit-shift operations are required for generating the predicted motion vector using (6). For each tested potential motion vector, 2 subtraction, 2 absolute value and 2 addition operations need to be performed for evaluating the modified cost function in (7). The complexity is still far less than that of SAD, which requires a subtraction, absolute value and addition operation for every pixel in the block, for every tested motion vector (e.g. using 8x8 blocks, 64 of each operation are required for every tested MV).

4. CONCLUSIONS

In this paper, a method is proposed for improving low complexity one bit or two bit transform based motion estimation through Motion Vector Prediction (MVP). In our method a predicted MV is formed for each block based on the MV’s of previous blocks. The cost function minimized to choose a MV is modified to penalize potential MV’s based on how much they differ from the predicted MV. Our proposed method results in a more uniform motion field and increases the subjective and objective quality of frames reconstructed through motion compensation.

5. REFERENCES

[1] T. Chen, T. Huang and L. Chen, "Analysis and design of macroblock pipelining for H.264/AVC VLSI architecture," in Proc. of the Int. Symposium on Circuits and Systems, ISCAS '04, 2004, pp. 273-76.

[2] J. Jain and A. Jain, “Displacement measurement and its application in internal image coding,” IEEE Trans. Commun., vol. 29, no. COM–12, pp. 1799–1808, Dec. 1981.

[3] S. Zhu and K.-K. Ma, “A new diamond search algorithm for fast blockmatching motion estimation,” IEEE Trans. Image Process., vol. 9, no. 2, pp. 287–290, Feb. 2000.

[4] J. Feng, K.-T. Lo, H. Mehrpour, and A. E. Karbowiak, “Adaptive block matching motion estimation algorithm using bit plane matching,” in Proc. ICIP’, 1995, pp. 496–499.

[5] M. M. Mizuki, U. Y. Desai, I. Masaki, and A. Chandrakasan, “A binary block matching architecture with reduced power consumption and silicon area requirement,” in Proc. IEEE ICASSP, vol. 6, Atlanta, GA, 1996, pp. 3248–3251.

[6] B. Natarajan, V. Bhaskaran, and K. Konstantinides, “Low-complexity block-based motion estimation via one-bit transforms,” IEEE Trans. Circuits Syst. Video Technol., vol. 7, no. 4, pp. 702–706, Aug. 1997.

[7] S. Erturk, “Multiplication-Free One-Bit Transform for Low-Complexity Block-Based Motion Estimation,” IEEE Signal Processing Letters, vol. 14, no. 2, Feb. 2007.

[8] A. Ertürk and S. Ertürk, “Two-bit transform for binary block motion estimation,” IEEE Trans. Circuit Syst. Video Technol., vol. 15, no. 7, pp. 938–946, Jul. 2005.

[9] Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec.H.264 jISO/IEC 14496-10 AVC), Joint Video Team, Mar. 2003, Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVT-G050.