

Low power variable block size motion estimation using pixel truncation

Asral Bahari, Tughrul Arslan and Ahmet T. Erdogan
School of Engineering and Electronics, University of Edinburgh
United Kingdom
Email: {a.bahari, tughrul.arslan, ahmet.erdogan}@ed.ac.uk

Abstract—This paper presents a method of low-power variable-block-size motion estimation using pixel truncation. Previous work focused on implementing pixel truncation using fixed-block-size motion estimation. However, pixel truncation fails to give satisfactory results for smaller block partitions. In this paper, we analyse the effect of truncating pixels for smaller block partitions and propose a method to improve the frame prediction. To further reduce power consumption, we adopt low-complexity matching criteria for the highly truncated bit. The low-complexity matching criteria can work together with pixel truncation to reduce computational complexity without significantly degrading picture quality.

I. INTRODUCTION

Video compression plays an important role in today's wireless communications. It allows raw video data to be compressed before it is sent through a wireless channel. This allows many multimedia applications such as video conferencing, video streaming and live TV broadcasting to be transmitted through portable multimedia devices.

However, video compression is computing intensive and dissipates a significant amount of power. This is a major limitation in today's portable devices. Existing multimedia devices can only play video applications for a few hours before the battery is depleted. This limits the user's experience and becomes a major bottleneck for the development of more attractive applications.

Motion estimation has been identified as the main source of power consumption in video encoders. It consumes 40% of the total power used in video compression. This results from the high computational load needed to predict the current frame. After finding the best match, the encoder sends the encoded difference between the actual and the predicted frame, rather than the original frame. The decoder will then reconstruct the frame based on the received information. Thus, the result at the receiver depends on how good the motion estimation prediction is.

Block-based motion estimation has been widely adopted by the industry due to its simplicity and ease of implementation. In block-base motion estimation, each frame is partitioned into smaller blocks known as macroblocks (16x16 pixels). Each macroblock in the current frame is predicted by finding the best match within a specified area in the previously encoded frames, as shown in Fig 1.

Pixel truncation can be used to reduce the computational load by allowing us to disable the hardware that processes

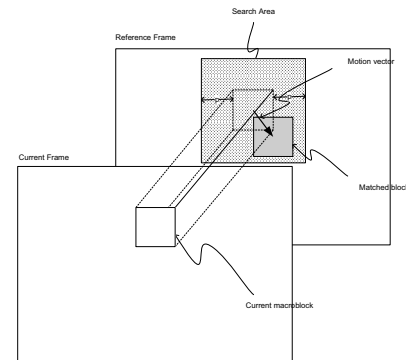


Fig. 1. Motion prediction using block base motion estimation

the truncated bits. In motion estimation, pixel truncation has been used to reduce the computational load for matching calculation unit [1]. The authors in [2] implemented adaptive pixel truncation on the least significant bits (LSB). The number of truncated pixels is modified according to the quantisation parameter (QP) value. Truncating the pixel's most significant bits (MSB) was discussed in [3].

While previous studies focused on fixed-block-size motion estimation (16x16 pixels), very little work has been done to study the effect of pixel truncation for smaller block sizes. The latest MPEG-4 standard, MPEG-4 AVC/H.264, allows variable block size for motion estimation (VBSME). [4] defines 16x16, 16x8, 8x16, 8x8, 8x4, 4x8 and 4x4 block size. At smaller block sizes, better prediction is achieved for objects with complex motion.

Truncating pixels at a 16x16 block size results in acceptable performance as shown in the literature [2]. However, at smaller block sizes, the number of pixels involved during motion prediction is reduced. Due to the truncation error, there is a tendency for smaller blocks to yield matched candidates which could lead to the wrong motion vector. Thus, truncating pixels at smaller block sizes results in poor prediction.

In this paper, we present a low-power method of motion estimation using pixel truncation for smaller block sizes. We perform the search in two steps: (1) truncation mode; and (2) refinement mode. This method allows us to truncate up to 6 LSB without significantly degrading the prediction accuracy. The rest of this paper is organised as follows. Section II discusses the pixel truncation for variable block size motion

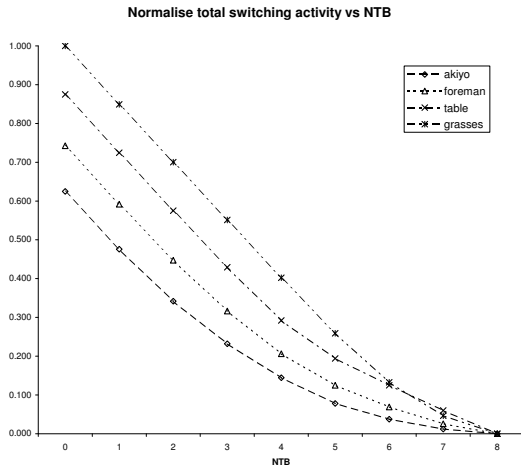


Fig. 2. Total switching activity vs NTB

TABLE I
CDF OF CALCULATED SAD DURING MOTION ESTIMATION

Block size	NTB	0	≤ 5%	≤ 10%	≤ 20%	≤ 40%
16x16	0	0%	25%	60%	94%	100%
	4	0.2%	25%	60%	94%	100%
8x8	0	0%	35%	58%	87%	98%
	4	5%	35%	58%	87%	98%
4x4	0	0%	58%	58%	81%	99%
	4	12%	58%	58%	81%	99%

estimation. Section III discusses our experimental results. Finally, Section IV concludes this paper.

II. PIXEL TRUNCATION FOR VBSME

For video application, the data is highly correlated and the switching activity distributed non-uniformly [3]. Since most of the switching activity concentrate at less significant bit (LSB), significant power reduction can be achieved by truncation these bits. Fig. 2 shows the total switching activity for different no of truncated LSB (NTB). In general, about 50% switching activity reduction is obtained if we truncate up to 3 LSB. Further reduction is achieved for NTB=6 where 80-90% switching activity is reduced. This makes pixel truncation attractive to minimise power in motion estimation.

From [2], pixels truncation for a 16x16 block size gives a small PSNR drop compared to untruncated pixels. At a 16x16 block size, there are 256 pixels involve in the matching calculation. Therefore, the probability of finding the best match close to full pixel resolution is high.

However, for smaller block sizes, the block contains fewer pixels for motion prediction. At a 4x4 block size, for example, there are only 16 pixels involved in the matching calculation. Therefore, during motion estimation using truncated bits, the small block size will give a larger prediction error.

Table I shows the Cumulative Distribution Function (CDF) for SAD that is obtained during motion estimation. The SAD is grouped into 5 categories: 0% represents the percentage for $SAD = 0$, 5% represents the percentage of $SAD <$

TABLE II
AVERAGE FULL SEARCH PSNR FOR VARIOUS NTB USING SAD AS
MATCHING CRITERIA (SEARCH RANGE $P = \pm 8$)

NTB	Block size					
	16x16	diff	8x8	diff	4x4	diff
0	33.11	0	34.89	0	36.82	0
1	33.12	0.01	34.85	-0.03	36.75	-0.07
2	33.12	0.01	34.85	-0.03	36.75	-0.07
3	33.11	0.00	34.75	-0.14	36.29	-0.53
4	33.03	-0.08	34.35	-0.54	34.66	-2.16
5	32.85	-0.26	33.21	-1.67	31.85	-4.97
6	31.79	-1.33	30.29	-4.60	27.46	-9.36
7	28.65	-4.46	24.82	-10.07	21.70	-15.12

$5\% SAD_{max}$ and so on. For 16x16 block size with NTB=4, the percentage of $SAD=0$ is closed to the untruncated bit (NTB=0). This shows that for 16x16 block size, the truncated pixel has more likely to have same matched candidate as in the untruncated pixel. However, for 4x4 block with NTB=4, the percentage of $SAD=0$ is 12% compared to 0% for NTB=0. This shows that there are more matched candidates using truncated pixel for 4x4 block size which could lead to incorrect motion vectors.

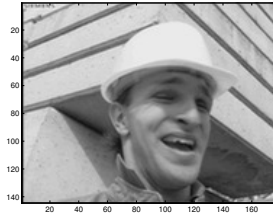
To illustrate the effect of pixel truncation on variable block size motion estimation, we calculate the average PSNR for 50 predicted frames of foreman sequence (QCIF@30fps) as shown in Table II. The frames are predicted using full search algorithm at different block sizes and numbers of truncated bits (NTB). A frame with three different NTB is shown in Fig. 3. From the table, for full pixel resolution (NTB = 0), the prediction accuracy improves as the block size decreases. This is reflected by a higher PSNR for prediction using a 4x4 block compared to a 16x16 block.

For NTB = 4, a small PSNR drop is observed for a block size of 16x16 (0.08 dB) compared to untruncated pixels. The PSNR drop for prediction using smaller block sizes is higher with 0.54 dB and 2.16 dB drops for frames with block sizes 8x8 and 4x4, respectively.

As we increase the NTB = 5, the PSNR drop for the smaller blocks increases rapidly. The PSNR drop for the 16x16 block size is 0.26 dB. However, for 8x8 and 4x4 block sizes, the PSNR drop increases to 1.68 dB and 4.97 dB, respectively. This shows that pixel truncation is not suitable for smaller block sizes, where the truncated blocks tend to find wrong motion vector compared to full pixel resolution. In the H.264 standard, substantial improvement in motion prediction is gained by using smaller blocks. Therefore, it is important to improve the PSNR gain especially for smaller block partitions.

In this paper, we propose a method of pixel truncation for variable-block-size motion estimation. This method is based on the following observations:

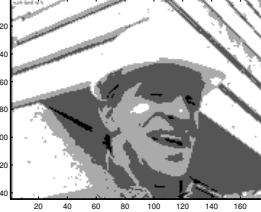
- 1) truncating pixels for larger block sizes can result in better motion prediction compared to smaller block sizes.



(a) NTB = 0



(b) NTB = 4



(c) NTB = 6

Fig. 3. Pixel truncation for foreman sequences

2) at higher pixel resolutions, smaller block sizes can result in better prediction compared to the larger block sizes.

To make sure that the large motion vector error can be avoided for smaller blocks, we implement the motion prediction in two steps as describe in Fig 4. In the first search, the prediction is performed using truncated bits with a 16x16 block size. Then, the result of the first search is refined using higher pixel resolution in a smaller search area.

The first search is important to identify the main direction of a macroblock movement. At the 16x16 block size, the truncated pixels contain enough information to enable the identification of the main feature of the block. We then compare it to the one in the search area using exhaustive search. Since in most cases, pixels within a macroblock move in a similar direction, we can obtain an initial estimate of the motion vector on the first search.

To achieve a more accurate motion vector, we use higher pixel resolution to refine the result obtained from the first search. The second search is performed within a smaller search area. This provides better accuracy and eliminates unnecessary motion vector errors for smaller blocks.

III. EXPERIMENTAL RESULTS

In this section, we first compare the PSNR results of the proposed method against conventional full search algorithm. For this comparison, we use SAD as the matching criteria. Next, we analyse the performance of error-based and boolean-based block matching criteria using truncated pixel. Finally, we present the result of the improved two-step search by combining the pixel truncation with low complexity matching criteria.

Table III shows the results of the proposed two-step search. SAD is used for matching criteria in the first and second search. In the first search, the search range $p = \pm 8$ is used with

```

// Truncating the search window, Y.
Yt = BITAND(Y, T)
// Truncating the current mb, X.
Xt = BITAND(X, T)
// Initialise mv and min.cost
mvx = 0, mvy = 0, cost = costmax
// Scanning the search windows and find the best match using block size N=16
For i = -p1, p1
  For j = -p1, p1
    For m = 1, N
      For n = 1, N
        cost = ∑p1i=-p1 ∑p1j=-p1 [MATCH1(Xt(m, n), Yt(i + m, j + n))]
      End of n
    End of m
    If(cost < costmin)
      costmin = cost, mvx = i, mvy = j
    End of j
  End of i
// Refining the search result using full pixel for variable block size
For i = -p2, p2
  For j = -p2, p2
    For m = 1, N
      For n = 1, N
        cost = ∑p2i=-p2 ∑p2j=-p2 MATCH2[X, Y]
      End of n
    End of m
    If(cost < costmin)
      costmin = cost, mvx = i, mvy = j
    End of j
  End of i

```

Fig. 4. Pixel truncation algorithm using two step approach

the NTB indicated by the first column. The second search is performed at full pixel resolution with a smaller search range with $p = \pm 4$.

From the table, we can see that using a two-step search improves the average PSNR for frame prediction up to 4x4 block size, compared to Table II. The PSNR drops well below 0.5 dB for NTB 4 and 5 compared to the untruncated pixel. This allows us to increase NTB without significantly degrading the picture quality. From a power consumption point of view, this method allows us to reduce power consumption by switching off the unused hardware.

On the other hand, for NTB greater than 5, the PSNR drop increases rapidly for 4x4 block size, with 0.85 dB and 3.51 dB for NTB 6 and 7, respectively. The error-based matching criteria such as SAD give poor results for very high NTB. Due to a large truncation error, the accumulated error during SAD calculation does not reflect the actual error of the matched block.

In order to improve prediction accuracy for NTB greater than 5, we several compared error-based matching criteria against boolean-based matching criteria. Two error-based matching criteria (MinMax [5] and MRMAD [6]) and two boolean-based matching criteria (BXOR [7] and DPC [8]) are compared against SAD.

Table IV shows the PSNR results for predicted frames using different matching criteria. The PSNR is calculated based on reconstructed frames during the first search. At NTB = 6 most error-based matching criteria result in low PSNR. For highly truncated bits, using boolean-based matching criteria is favourable compared to SAD. Among the compared matching criteria, DPC shows the best PSNR for NTB = 6, with 0.69 dB drop. This gives us a good estimate of the macroblock

TABLE III

AVERAGE PSNR COMPARISON FOR THE TWO-STEP SEARCH USING SAD AS MATCHING CRITERIA IN THE FIRST AND SECOND SEARCH.

NTB	Block size					
	16x16	diff	8x8	diff	4x4	diff
0	33.11	0	34.89	0	36.82	0
1	32.73	-0.38	34.33	-0.56	36.02	-0.8
2	33.12	0.01	34.77	-0.12	36.50	-0.32
3	33.13	0.02	34.77	-0.12	36.50	-0.32
4	33.10	-0.01	34.78	-0.11	36.50	-0.32
5	33.07	-0.04	34.73	-0.16	36.45	-0.37
6	32.66	-0.45	34.26	-0.63	35.97	-0.85
7	30.66	-2.45	31.92	-2.97	33.31	-3.51

TABLE IV

AVERAGE PSNR FOR DIFFERENT BLOCK MATCHING CRITERIA USING 16X16 BLOCK SIZE

NTB	Matching criteria				
	SAD	MRMAD	MinMax	BXOR	DPC
4	33.03	33.01	32.24	32.24	31.67
6	31.79	32.25	32.17	32.17	32.42

motion direction which will be refined in the second search. In addition, compared to SAD, DPC can be implemented with lower computational cost by using XOR-gates.

To minimise the power consumption, we proposed the modified two-step search by implementing the two-step search using DPC and SAD as matching criteria. The DPC is used as matching criteria during the first search, while in the second search, the refinement is done using SAD. Table V shows the results of this implementation. From the table, the results for NTB greater than 5 improves compared to results in Table III. In this approach, only 0.41 dB drop is observed for prediction using 4x4 block size compared to untruncated pixel for NTB = 6. From visual inspection point of view, this results in minimal drop in picture quality, as shown in Fig. 5.

The propose method allows us to trade-off the computational load without significantly reduce the prediction accuracy. Even though an additional step and small increases in area is required to ensure high prediction accuracy especially for smaller block size, the overall computational load is still low compared to conventional full search algorithm. This is achieved by using low computational cost matching criteria in the first search and smaller search window during refinement step.

IV. CONCLUSION

This paper discusses a method of low-power variable-block-size motion estimation using pixel truncation. Previous work has shown that pixel truncation results in acceptable performance for motion prediction using a 16x16 block size. However, for motion prediction using smaller block sizes, pixel truncation reduces the motion prediction accuracy. In

TABLE V

AVERAGE PSNR COMPARISON FOR THE IMPROVE TWO-STEP SEARCH USING DPC IN THE FIRST SEARCH AND SAD IN THE SECOND SEARCH

NTB	Block size					
	16x16	diff	8x8	diff	4x4	diff
4	33.09	-0.02	34.76	-0.13	36.49	-0.33
5	33.09	-0.02	34.75	-0.14	36.48	-0.34
6	33.03	-0.08	34.69	-0.20	36.41	-0.41
7	32.79	-0.32	34.43	-0.46	36.12	-0.7



Fig. 5. Frame predicted using the Two-step method for NTB=6

this paper, we propose a two-step search to improve frame prediction using pixel truncation. In addition, we combine DPC and SAD during motion estimation to further improve the prediction accuracy and reduce the computational complexity. Our method shows good PNSR for NTB up to 6 without significantly degrade the picture quality.

REFERENCES

- [1] Yin Chan and S.Y. Kung, "Multi-level pixel difference classification methods", *IEEE International Conference on Image Processing*, vol. 3, pp. 252 – 255, 1995.
- [2] Zhong-Li He, Chi-Ying Tsui, Kai-Keung Chan, and Ming L. Liou, "Low-power vlsi design for motion estimation using adaptive pixel truncation", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, pp. 669 – 678, 2000.
- [3] Vasily G. Moshnyaga, "Msb truncation scheme for low-power video processors", *Proceedings - IEEE International Symposium on Circuits and Systems*, vol. 4, pp. 291–294 –, 1999.
- [4] "Itu-t recommendation h.264 & iso/iec 14496-10 (mpeg-4) avc, "advanced video coding for generic audiovisual services"", 2005.
- [5] Mei-Juan Chen, Liang-Gee Chen, Tzi-Dar Chiueh, and Yung-Pin Lee, "A new block-matching criterion for motion estimation and its implementation", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 5, no. 3, pp. 231 – 6, June 1995.
- [6] S. Sharma, P. Mishra, S. Sawant, C.P. Mammen, and V.M. Gadre, "Pre-decision strategy for coded/non-coded mbs in mpeg4", *2004 International Conference on Signal Processing and Communications (SPCOM)*, pp. 501 – 5, 2004.
- [7] Hangu Yeo and Yu Hen Hu, "A novel architecture and processor-level design based on a new matching criterion for video compression", *VLSI Signal Processing*, IX, pp. 448 – 57, 1996.
- [8] Seongsoo Lee, Jeong-Min Kim, and Suo-Ik Chae, "New motion estimation algorithm using adaptively quantized low bit-resolution image and its vlsi architecture for mpeg2 video encoding", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 6, pp. 734 – 44, Oct 1998.