Fast DC Mode Prediction Scheme For Intra 4x4 Block In H.264/AVC Video Coding Standard

Tajdid Ul Alam[1]
Department of Electronics and Telecommunication Engineering
Rajshahi University of Engineering and Technology, Rajshahi, Bangladesh

Jafor Ikbal[2]
Department of Electronics and Telecommunication
Engineering Rajshahi University of Engineering and
Technology, Rajshahi, Bangladesh

Touhid Ul Alam[3]
Department of Computer Science and Engineering,
Ahsanullah University of Science and
Technology, Dhaka, Bangladesh

Abstract— **In this paper, the researchers proposed a new scheme for DC mode prediction in intra frame for 4X4 block. In this scheme, the upper and left neighboring pixels would be used to predict each of the pixels in the 4X4 block with different weight. The prediction equations for each position of the 4X4 block in DC mode would be static with fixed weighting coefficients. As a result, the computational time for intra frame would be decreased considerably with an increase in PSNR.**

Keywords- Intra; macroblock (MB); DC mode; luma; prediction.

I. Introduction

H.264 or MPEG-4 Part 10 Advanced Video Coding (AVC) is a joint venture of video coding experts from both the Joint Video Team (JVT) of ITU-T and Motion Picture Experts Group (MPEG) of ISO [1].

H.264/AVC is an important format, which is most commonly used in recording, compression and distribution of high definition video. It was first released in May, 2003.

After its release, it has been found that it has very broad application that covers all forms of digital compressed video from low bit-rate internet streaming applications to HDTV broadcast and digital cinema applications where nearly lossless coding is very important [2].

A number of new features have been added to the H.264/AVC. Some of these features are multiple reference frames, variable block size motion estimation, multiple motion vectors per macroblock, weighted prediction, spatial prediction from the edges of neighboring block for intra coding rather than DC only etc. These features allow H.264/AVC to compress video more efficiently but complexity of implementation also increases.

In H.264/AVC standard, spatial redundancy is removed in intra frame by using 9 modes [3]. DC mode is one of the modes that are used for encoding intra 4X4 block.

In DC mode, the luminance values of entire pixels of current 4X4 block are predicted by computing average of the luminance values of pixels from upper and left neighbors. For 4X4 block, there are 8 such pixels in total.

We observed that most of the variations of luminance values in intra 4X4 block are removed by the standard DC mode as the luminance value of 16 pixels are predicted by a single value, which is calculated by averaging the 8 pixels. Therefore, we are enthusiastic to propose a new pattern for DC mode which will preserve the variations of luminance values. We call this pattern as “ZIG-ZAG DC mode prediction”.

A lot of efforts have been made to improve the intra prediction scheme of H.264/AVC. Many researches have been done to reduce computational complexity. But it has been witnessed that in most of the cases the researchers have less contribution in improving coding efficiency.

This ZIG-ZAG DC mode prediction can improve coding efficiency by preserving more details and can reduces computational time with significant increase in PSNR.

II. Intra Prediction Scheme In H.264/Avc Standard

Prediction for intra macroblocks (MB) is called intra prediction. Intra prediction is used to remove spatial redundancy of intra frames. Intra frames are encoded without any reference of other frames. Intra prediction is performed in pixel domain. It uses only transform coding and the neighboring blocks of the same frame to predict block values. Intra prediction is performed on 16X16 luma and 4X4 luma blocks.

No intra prediction is performed on 8X8 luma blocks [4]. In this standard, intra prediction forms predictions of pixel values as linear interpolations of pixels from the adjacent edges of neighboring MBs that are decoded before the current MB that is to be encoded. The interpolations are directional in nature, with multiple modes, each implying a spatial direction of prediction [3].

A 4X4 luma block contains samples as shown in figure 1 (a). There are nine intra prediction modes as shown in figure 1 (b). The modes are given in table 1.

TABLE 1: INTRA PREDICTION MODES FOR 4X4 LUMA BLOCK.

Number	Intra 4X4 prediction mode
0	Vertical
1	Horizontal
2	DC
3	Diagonal down left
4	Diagonal down right
5	Vertical right
6	Horizontal down
7	Vertical left
8	Horizontal up

Modes 0, 1, 3, 4, 5, 6, 7, 8 are directional prediction modes as indicated in figure 1 (b) and mode 2 is the DC prediction mode with no direction. The prediction block is calculated from the samples A-M as shown in figure 1 (a). In figure 1 (b), the arrows indicate the direction of prediction in each mode.

While predicting a sample in the current 4X4 block, the neighboring samples of upper side and left side have previously been encoded and reconstructed. The neighboring samples are available to the encoder and decoder to form prediction references.

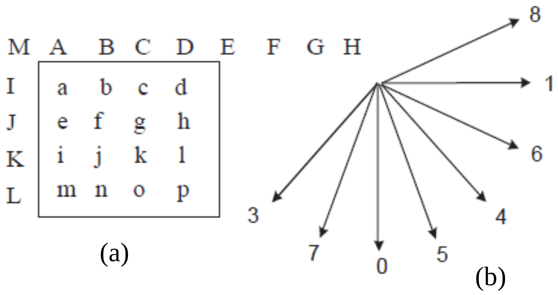


Figure 1: (a) Identification of samples used for intra spatıaı prediction, (b) intra prediction directions.

I. Standard Algorithm For Dc Mode Prediction

The standard form of DC mode prediction for 4X4 block is to replace all pixels in the current 4X4 block by the mean value of the neighboring pixels. The pixels in the current 4X4 block that are to be predicted are shown in figure 1(a) by using a box. The reference pixels are A, B, C, D, I, J, K and L. These are also shown in the same figure. The DC prediction is accomplished by using the following rules [4]:

1. If all samples A, B, C, D, I, J, K, L are available, all samples are predicted by (A+B+C+D+I+J+K+L+4)>>3
2. If A, B, C, D are not available and I, J, K, L are available, all samples shall be predicted by (I+J+K+L+2)>>2
3. If I, J, K, L are not available and A, B, C, D are available, all samples shall be predicted by (A+B+C+D+2)>>2
4. If all eight samples are unavailable, the prediction for all luma samples in the 4X4 block shall be 128.

A block therefore can always be predicted in DC mode. A typical case for luma prediction using DC mode in 4X4 block is shown in figure 2 [5].

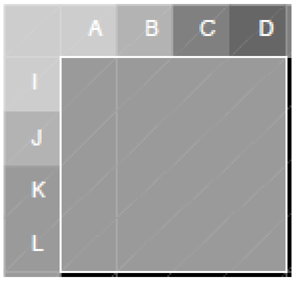


Figure 2: A typical case of DC Mode prediction.

Figure 2 shows that all pixels in the current 4X4 block are predicted by an average of their neighboring pixels. So, most of the variations of luminance value are removed by using DC prediction mode.

III. Proposed Algorithm For Dc Mode Prediction

The DC mode prediction scheme of pixel values in current 4X4 block can be improved by using ZIG-ZAG DC mode prediction. In ZIG-ZAG DC mode prediction, the 16 pixels in current 4X4 block are predicted one by one using a ZIG-ZAG order. The prediction sequence of the pixels in current 4X4 block is indicated in figure 3 using arrows. The reference pixels are A, B, C, D which belong to the upper neighboring block and I, J, K, L which belong to the left neighboring block. Besides these reference pixels, we also used previously predicted pixels of the current 4X4 block to predict another pixel belonging to the same block. But the priority of pixels from upper and left neighboring block was high and priority of pixels belonging to the same block was low. This was done by assigning different weighting coefficient to different pixels.

For example, we use predicted luminance values of pixels in the positions of i, e, f and g to predict the luminance value of pixel in the position of j. As shown in the figure, i, e, f and g are previously predicted pixels positions. Similarly, luminance value of a pixel in the position of c is predicted from the predicted luminance value of pixels in the positions of f, b, B, C and D.

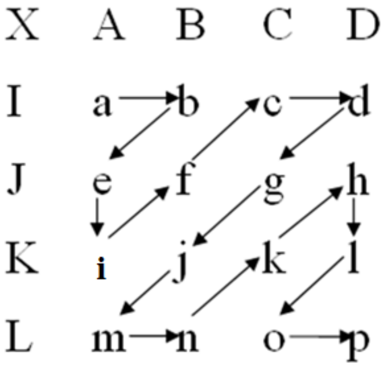


Figure 3: ZIG-ZAG DC mode prediction

Each pixel in current 4X4 block was predicted using different prediction equations. Each equation contains the luminance value of the pixels that are used to predict the luminance value of the current pixel in 4X4 block. The weight assigned to each of the reference pixels was different and it also depended on the pixel's position in current 4X4 block. We assigned higher weight to the upper and left neighboring pixels than the pixels belonging to the current 4X4 block.

The equations that are used for the prediction of luminance value of each pixel in the 4X4 block in ZIG-ZAG DC mode are given as follows:

$$a = 0.25A + 0.125B + 0.25X + 0.25I + 0.125J$$

$$b = 0.25A + 0.25B + 0.0833X + 0.0833I + 0.0833J + 0.25C$$

$$e = 0.05A + 0.05B + 0.3K + 0.3I + 0.3J$$

$$i = 0.06A + 0.06B + 0.06X + 0.2I + 0.2J + 0.2K + 0.2L$$

$$f = \frac{a + b + e + i}{4}$$

$$c = \frac{(B + C + D) + b + f}{3}$$

$$d = \frac{c + (C + D)}{2}$$

$$g = \frac{b + c + d + f}{4}$$

$$j = \frac{e + f + g + i}{4}$$

$$m = \frac{i + j + (K + L)}{3}$$

$$n = \frac{i + j + k}{3}$$

$$k = \frac{f + g + j + n}{4}$$

$$h = \frac{k + j + c + d}{4}$$

$$l = \frac{g + h + k}{3}$$

$$o = \frac{j + k + l + n}{4}$$

$$p = \frac{o + k + l}{3}$$

We can see from the equations that the pixels in current 4X4 block are predicted using both the previously encoded pixels of upper and left neighbor and previously predicted pixels of the same block with different weighting coefficients.

From these equations we can also tell that, the luminance values of each position of the current 4X4 block are not same. So, the proposed scheme preserves more details than the previous scheme which leads to an increase in PSNR.

IV. SIMULATION RESULT

The proposed algorithm was coded and simulated in JM software version 18.2 [http://iphome.hhi.de/suehring/tml/]. Using the reference software, first we encoded the 10 sequences with the standard DC mode prediction algorithm and collected the PSNR(Y), computational time and bit rate. Then we coded and simulated the ZIG-ZAG DC mode prediction algorithm in JM software version 18.2 and then encoded the same sequences with the proposed ZIG-ZAG DC mode prediction algorithm and collected the PSNR(Y), computational time and bit rate of those sequences.

We encoded three frames for each sequences and all sequences were encoded in QCIF resolution. We compared the PSNR(Y), computational time and bit rate of the encoded sequences for both the standard DC mode prediction algorithm and ZIG-ZAG DC mode prediction algorithm. We observed the improvement of PSNR(Y) and computational time for most of the sequences where the bit rates for the sequences were almost the same.

The difference in PSNR(Y), computation time and bit rate between standard algorithm and proposed ZIG-ZAG DC mode prediction algorithm was calculated. The machine was Intel® Core™ i5-2430M @ 2.40GHz with 4.00GB RAM. Operating system was 64 bit OS. The condition for testing of standard algorithm and proposed algorithm were same. Observed change in results was tabulated as follows:

TABLE 2: SIMULATION RESULTS OF VARIOUS SEQUENCES.

Sequence	ΔPSNR (dB)	Δ bit rate (kbps)	Δ time (ms)
Miss America	0.048	1.44	0.361
Bus	-0.009	-20.16	-14.725
Foreman	1.588	-0.12	-12.372
Mobile	-0.095	9.96	-12.53
Mother	2.679	2.76	0.057
Highway	0.066	2.28	-8.594
Coastguard	0.119	8.28	-0.777
Grandma	0.059	1.68	-0.008
Salesman	0.417	-3.60	-68.291
Paris	0	0	-9.627
Average	0.4872	0.252	-12.651

In table 2, we calculated the differences of the PSNR(Y), computational time and bit rate of the standard algorithm from the ZIG-ZAG DC mode prediction algorithm.

As we can see from table 2, the PSNR(Y) increases in most of the cases. For the sequence "Paris", the PSNR(Y) was same. For the sequences "Bus" and "Mobile", the decrease in PSNR(Y) for ZIG-ZAG DC mode prediction algorithm is almost negligible.

Also we observe that, the decrease in computational time for the ZIG-ZAG DC mode prediction algorithm is noticeable for most of the sequences. For the sequences “Miss America” and “Mother-daughter” the difference in computational time is almost the same.

We calculated the average of the change in PSNR(Y), computational time and bit rate. From the average value, we observed that, increase in PSNR(Y) and computational time were satisfactory. The change in bit rate was almost negligible.

Our target was to increase the PSNR(Y) value and decrease the computational time. As we observed that the standard algorithm for DC mode prediction looks for several conditions for predicting each 4X4 block, the computation time for standard DC mode prediction algorithm is higher. The conditions are whether the upper samples are available or not, whether the left samples are available or not, whether the upper and left samples are available together, whether there is no sample available.

These four steps consume most of the time for predicting the pixel values in DC mode. As in our proposed ZIG-ZAG DC mode prediction algorithm, there is no such condition to check for, the computational time decreases than the standard DC mode prediction algorithm.

Again in standard DC mode prediction algorithm, all pixels in current 4X4 block are predicted by a single value as discussed earlier. But in our proposed ZIG-ZAG DC mode prediction algorithm, each pixel is predicted separately and variation in luminance values are preserved more in current 4X4 block. As a result for most of the cases, the PSNR(Y) increases.

Figure 4 to 6 shows the graph of PSNR(Y), computation time and bit rate values respectively. Each graph contains values for 10 sequences. The result for standard algorithm and proposed algorithm are shown side by side on same graph.

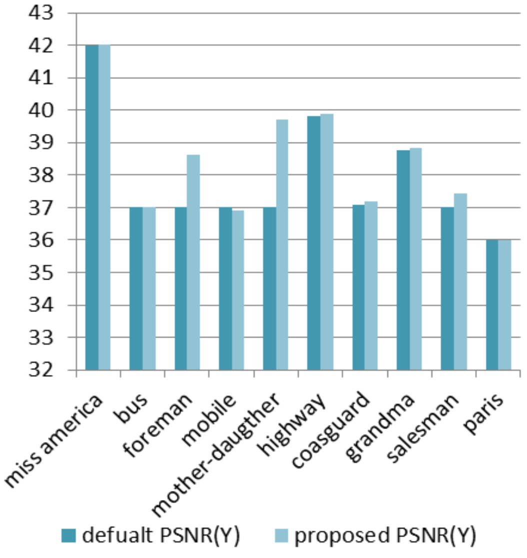
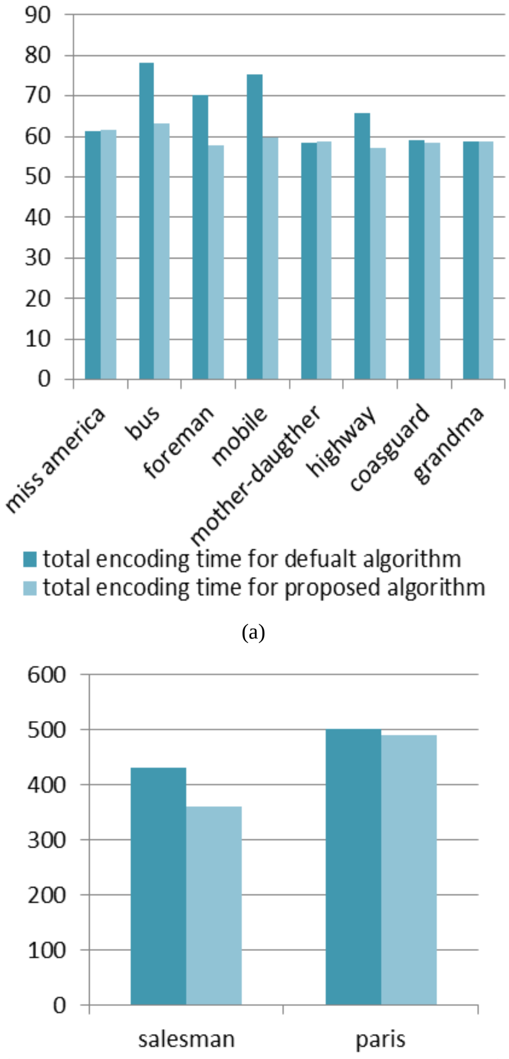


Figure 4: Comparison of PSNR(Y) of standard and ZIG-ZAG DC mode prediction algorithm.



(a)

(b)

Figure 5 (a) & (b): Comparison of total encoding time of standard and ZIG-ZAG DC mode prediction algorithm.

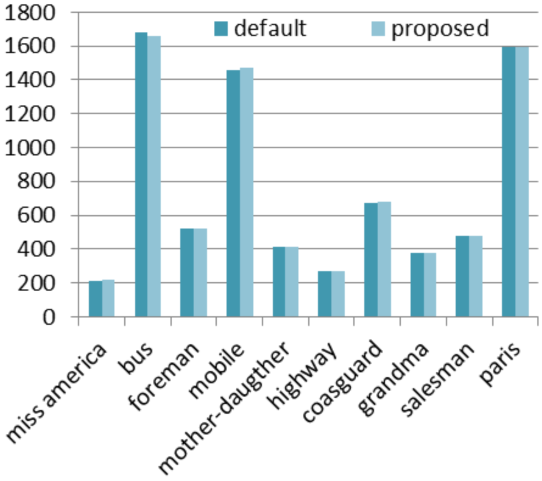


Figure 6: Comparison of bit rate (kbps) of standard and ZIG-ZAG DC mode prediction algorithm.

In standard DC mode, due to the replacement of all the pixels with a single value, most of the variations in luminance were omitted. But with our proposed ZIG-ZAG DC mode prediction algorithm, this factor is improved and as a result the PSNR increases. The decrease in processing time with the ZIG-ZAG DC mode prediction algorithm is due to the absence of conditions as involved with the standard DC mode prediction algorithm.

Actually in ZIG-ZAG DC mode prediction algorithm, the concepts of all the eight modes of prediction are combined. As a result, the prediction accuracy is improved.

V. CONCLUSION

From the observation and comparison of simulation results, we see that the computation time decrease which makes the ZIG-ZAG DC mode prediction algorithm more efficient for live video broadcasting application as well as teleconferencing application where lowering computation time adds advantage. Also increase in PSNR improves signal quality for the video signal.

VI. FUTURE WORK

In future, we will try to further decrease the computation time for intra frame in H.264/AVC standard. We can do this by observing the relative usage of each of the nine modes in intra frame and discarding the less used modes for prediction of intra frame. If the changes in PSNR(Y) and bit rate were not worse, then the reduction in prediction mode in intra frame would reduce the computation time.

REFERENCES

[1] Atul Puri, Xuemin Chen, Ajay Luthra, “Video coding using the H.264/MPEG-4 AVC compression standard”, Elsevier: Signal Processing: Image Communication 19 (2004) 793–849

[2] http://en.wikipedia.org/wiki/H.264/MPEG-4_AVC

[3] Soon-kak Kwon, A. Tamhankar, K.R. Rao, “Overview of H.264/MPEG-4 part 10”, science direct, J. Vis. Commun. Image R. 17 (2006) 186–216

[4] A. Tamhankar and K. R. Rao, “AN OVERVIEW OF H.264 / MPEG4 PART 10”, EC-VIP-MC 2003,4th EURASIP Conference focused on Video / Image Processing and Multimedia Communications, 2-5 July 2003, Zagreb. Croatia

[5] Gary J. Sullivan, Pankaj Topiwala, and Ajay Luthra, “The H.264/AVC Advanced Video Coding Standard:Overview and Introduction to the Fidelity Range Extensions”, Gary J. Sullivan, Pankaj Topiwala, and Ajay Luthra, Presented at the SPIE Conference on Applications of Digital Image Processing XXVII, Special Session on Advances in the New Emerging Standard: H.264/AVC, August, 2004

[6] JM Software: http://iphome.hhi.de/suehring/tml/

[7] Iain Richardson, “White Paper: H.264 / AVC Intra Prediction”, Vcodex, © 2002-2011

[8] Thomas Wiegand, Gary J. Sullivan, Gisle Bjøntegaard, and Ajay Luthra, “Overview of the H.264/AVC Video Coding Standard”, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL. 13, NO. 7, JULY 2003

[9] Yung-Lyul Lee, Ki-Hun Han, Dong-Gyu Sim, and Jeongil Seo, "Adaptive Scanning for H.264/AVC Intra Coding",aETRI Journal, Volume 28, Number 5, October 2006

[10] ITU-T Recommendation H.264 and ISO/IEC 14496-10, “Advanced video coding for generic audiovisual services,” May 2003 (and subsequent amendment and corrigenda).

[11] Wes Simpson, "Video Over IP IPTV, Internet Video, H.264, P2P, Web TV, and Streaming: A Complete Guide to Understanding the Technology",Second Edition, Published by Elsevier Inc.

[12] L. Hanzo, P. J. Cherriman and J. Streit, "Video Compression and Communications From Basics to H.261, H.263, H.264, MPEG4 for DVB and HSDPA-Style Adaptive Turbo-Transceivers, Second Edition, John Wiley & Sons, Ltd

[13] Mohammed Ebrahim Al-Mualla, C. Nishan Canagarajah and David R. Bull, "Video Coding for Mobile Communications Efficiency Complexity and Resilience", ACADEMIC PRESS

[14] John Watkinson, "The MPEG Handbook—MPEG-1, MPEG-2, MPEG-4", Focal Press

[15] Atul Puri, Tsuhan Chen, "Multimedia Systems, Standards, and Networks", MARCEL DEKKER, INC.