A Low – Power VLSI Architecture for Intra Prediction in H.264

Georgios Stamoulis[1], Maria Koziri[1], Ioannis Katsavounidis[2], and Nikolaos Bellas[3]

[1] University of Thessaly, Department of Computer and Communication Engineering, Glavani 37, 382 21 Bolos, Greece
{georges, mkoziri}@uth.gr
[2] InterVideo, Inc., 46430 Fremont Blvd., Fremont, CA 94538, USA
ioannis@intervideo.com
[3] Motorola, Inc., 1030 E. Algonquin. Rd, IL01-Annex A, Schaumburg, IL 60196, USA
bellas@labs.mot.com

Abstract. The H.264 video coding standard can achieve considerably higher coding efficiency than previous standards. The key to this high code efficiency are mainly the Intra and Inter prediction modes provided by the standard. However, the compression efficiency of the H264 standard comes at the cost of increased complexity of the encoder. Therefore it is very important to design video architectures that minimize the cost of the prediction modes in terms of area, power dissipation and design complexity. A common aspect of the Inter and Intra Prediction modes, is the Sum of Absolute Differences (SAD). In this paper we present a new algorithm that can replace the SAD in Intra Prediction, and which provides a more efficient hardware implementation.

1 Introduction

Video has always been the backbone of multimedia technology. In the last two decades, the field of video coding has been revolutionized by the advent of various standards like MPEG-1 to MPEG-4 and H.261 to H.263, each addressing different aspects of multimedia. The H.264 standard [1] pushes the envelope of video compression efficiency and provides a complete solution for a wide range of applications. The standard is being developed by the Joint Video Team(JVT) from the ISO/IEC and ITU-T. The primary goal of H.264 is to achieve higher compression while preserving video quality. The motivation of compression is to compensate for the ever present constraints of the limited channel capacity. This video coding technique follows a straight –forward "back to basics approach" providing flexibility to be used in low-delay real-time applications.

Each picture of a video, which can either be a frame or a field, is partitioned into fixed-size macroblocks that cover a rectangular picture area of 16×16 samples of the luma component and 8×8 samples of each of the two chroma components. All luma and chroma samples of a macroblock are either spatially or temporally predicted, and the resulting prediction residual is transmitted using transform coding. Therefore, each color component of the prediction residual is subdivided into blocks. The mac-

P. Bozanis and E.N. Houstis (Eds.): PCI 2005, LNCS 3746, pp. 633 – 640, 2005.

roblocks are organized in slices, which generally represent subsets of a given picture that can be decoded independently. The H.264/AVC standard supports five different slice-coding types. The simplest one is the I slice (where "I" stands for intra). In I slices, all macroblocks are coded without referring to other pictures within the video sequence. On the other hand, prior-coded images can be used to form a prediction signal for macroblocks of the predictive-coded P and B slices (where "P" stands for predictive and "B" stands for bi-predictive). The remaining two slice types are SP (switching P) and SI (switching I), which are specified for efficient switching between bitstreams coded at various bit-rates.

The H.264 standard uses block sizes of 4x4 and 16x16 pixels to compress I-Macroblocks for intra-prediction. Intra coding refers to the case where only spatial redundancies within a video picture are exploited. The resulting frame is referred to as an I-picture. I-pictures are typically encoded by directly applying spatial transform to the different macroblocks in the frame. As a consequence, encoded I-pictures are large in size since a large amount of information is usually present in the frame, and no temporal information is used as part of the encoding process. In order to increase the efficiency of the intra coding process in H.264, spatial correlation between adjacent macroblocks in a given frame is exploited. The idea is based on the observation that adjacent macroblocks tend to have similar properties. Therefore, as a first step in the encoding process for a given macroblock, one may predict the macroblock of interest from the surrounding macroblocks, typically the ones located on top and to the left of the macroblock of interest, since those macroblocks would have already been encoded. After a prediction block P is formed based on previously encoded and reconstructed blocks, it is subtracted from the current block prior to encoding. For the luma samples, the P block is formed for each 4x4 block or for a 16x16 macroblock. There are a total of nine optional prediction modes for each 4x4 luma block, four modes for a 16x16 luma block and four modes for the chroma components. The encoder typically selects the prediction mode for each block that minimises the difference between P and the block to be encoded. The selection is done by using SAD which indicates the magnitude of the absolute error.

In the context of hardware implementation, the calculation of SAD for each block requires a significant number of additions. In an attempt to reduce encoder's complexity and power consumption, this paper introduces a new technique that replaces SAD, without having major effects in the quality of the encoded image and time required for encoding. This technique can easily be translated to a simple yet effective low power VLSI architecture for H.264 Intra Prediction. Based on the concept of comparison, the proposed architecture, instead of adding the differences between the predicted and the original pixels and then comparing the resultant sums, compares these differences. At the end of the comparisons the prediction mode with the minimum difference is selected. This new architecture results in a much more simple circuit, than that required for the complete calculation of SAD. Therefore, significantly reduction of the time required for Intra Prediction and low-power consumption will be achieved. In section 2 of the paper, we outline the H.264 intra prediction mode, and in section 3 we discuss our algorithm and explain how it reduces the computational complexity of the intra prediction mode. In section 4, we present the experimental evaluation of our algorithm, and we conclude in section 5.

2 Intra Prediction for H.264

The H.264 standard exploits the spatial correlation between adjacent macroblocks/blocks for Intra prediction. That is, the current macroblock/blocks is predicted using adjacent pixels in the upper and the left macroblocks/blocks that are decoded earlier. The H.264 standard offers a rich set of prediction patterns for Intra prediction, *i.e.* nine prediction modes for 4x4 luma blocks and four prediction modes for 16x16 luma blocks. Each mode has its own direction of prediction and the predicted samples are obtained from a weighted average of decoded values of neighbourhood macroblocks/blocks [2].

Suppose that we have a 4x4 luma block that is required to be predicted. The samples above and to the left, which are labelled A–M, have previously been encoded and reconstructed and are therefore available in the encoder and decoder to form a prediction reference. The samples a, b, c, . . . , p of the prediction block P (Figure 1) are calculated based on the samples A–M as follows. Mode 2 (DC prediction) is modified depending on which samples A–M have previously been coded; each of the other modes may only be used if all of the required prediction samples are available. Note that if samples E, F, G and H have not yet been decoded, the value of sample D is copied to these positions and they are marked as 'available'.

M	A	B	C	D	E	F	G	H
I	a	b	c	d				
J	e	f	g	h				
K	i	j	k	l				
L	m	n	o	p				

Fig. 1. Labeling of prediction samples

In Figure 2 we can see a schematic representation of the nine available modes for intra prediction of a 4x4 block of luma samples. Table 1 gives us more details about how the nine modes predict the 4x4 block.

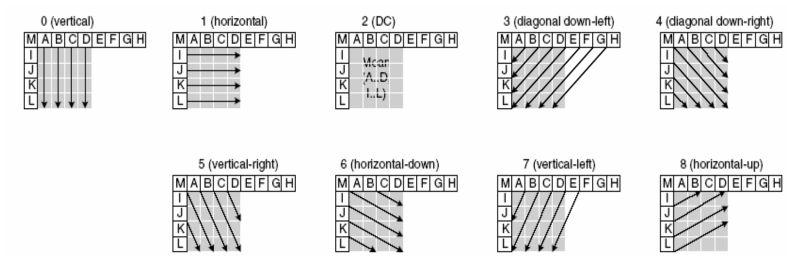

Fig. 2. 4x4 luma prediction modes

Table 1. The nine 4x4 luma block Intra prediction modes in H.264

Mode 0 (Vertical)	The upper samples A, B, C, D are extrapolated vertically.
Mode 1 (Horizontal)	The left samples I, J, K, L are extrapolated horizontally.
Mode 2 (DC)	All samples in P are predicted by the mean of samples A . . . D and I . . . L.
Mode 3 (Diagonal Down-Left)	The samples are interpolated at a 45· angle between lower-left
Mode 4 (Diagonal Down-Right)	The samples are extrapolated at a 45· angle down and to the right.
Mode 5 (Vertical-Right)	Extrapolation at an angle of approximately 26.6· to the left of vertical (width/height = 1/2).
Mode 6 (Horizontal-Down)	Extrapolation at an angle of approximately 26.6· below horizontal.
Mode 7 (Vertical-Left)	Extrapolation (or interpolation) at an angle of approximately 26.6· to the right of vertical.
Mode 8 (Horizontal-Up)	Interpolation at an angle of approximately 26.6· above horizontal.

To decide which mode will be selected the Sum of Absolute Differences (SAD) is calculated for each mode. Intuitively speaking, a good prediction should produce a small value of the sum of absolute differences (SAD), which can be written as

$$SAD = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |C_{ij} - P_{ij}| \tag{1}$$

where C and P represent the current block and its prediction, respectively[3]. According to these, the mode which has the minimum SAD is the one to be selected. The SAD computations are very expensive because they require a large number of computations for each 4x4 block. Going down to hardware level the cost for this is not negligible.

3 Proposed Intra-Mode Decision Algorithm

In this section we introduce a new technique for approaching the problem of Intra – Mode decision. The base of this technique is to avoid the stage of addition, which augments significantly the cost in the hardware level.

For a given 4x4 block, according to equation (1), a total of 16 subtractions and 15 additions is needed in order to produce the SAD for one mode. Therefore for the nine modes we conclude to 144 subtraction and 135 additions. After computing all modes, a comparison between the results will give us the decided mode. Since all that is needed in this stage of Intra Prediction is to find the prediction mode, a qualitative approach may give the same results as a quantitative one.

Based on the above observation, the proposal is, after calculating the differences among the predicted and the original pixels instead of adding them, to compare the differences for the nine available modes. The comparison will conclude to the mode with the most minimum differences. In this way, the addition stage is completely bypassed.

We first calculate the absolute difference between the corresponding pixels for each mode. This can be written as

$$M_{k_{ij}} = | C_{ij} - P_{k_{ij}} | \tag{2}$$

where M, C, P are 4x4 arrays and k (with $0 \leq k \leq 8$) indicates the mode.

After the nine 4x4 arrays are created, a comparison among two successive arrays is done. The array with the largest number of minimum values is chosen. Lets assume that we have the following function,

$$F_k = \begin{cases} 1, M_k \leq M_{k+1} \\ 0, M_k > M_{k+1} \end{cases} \tag{3}$$

where M_k is the array with the differences for mode k. According to equation (3) we chose M_k if $F_k < F_{k+1}$, otherwise we chose M_{k+1}.

The above algorithm may be more easily understood if presented with a code-like form.

```
for (k=0;k<NO_INTRA_PMODE;k++){
k1=0;
            counter1=0;
            counter2=0;
            for (j=0;j<4;j++) {
              for (i=0;i<4;i++) {
                M[k][i][j] = abs(C[i][j] - P[k][i][j]);
                k1=k+1;
                M[k1][i][j] = abs(C[i][j] - P[k1][i][j]);
                if (M[k][i][j]>= M[k1][i][j])
                  Mode[next] = k;
                else
                  Mode[next] = k+1;
              }
            }
            k=k1;
            next++;
}
```

A total of 8 comparisons can produce the mode with the least number of differences from the original 9 candidates.

Therefore for the proposed algorithm we have a total of 160 comparisons instead of 135 additions and 8 comparisons which are needed with the use of SAD. Nevertheless, when thinking about a hardware implementation, these results are quite inviting, especially if one takes into account the fact that the comparison is a stage which is also met into an implementation with SAD.

4 Experimental Results

The proposed mode decision scheme has been integrated with the H.264 JM9.3 codec for the performance evaluation. It is compared with the Intra mode decision of H.264 (with SAD) in terms of the size of the file produced after the total encoding process and the average PSNR as a function of the frames encoded and as a function of the quantization parameter QP for the foreman test sequence recommended in [4]. Fig. 3 shows the % difference of the size of the files produced by the encoder for the foreman sequence of various number of frames and various QPs at coding rate 15 frames/sec. As one can notice the new algorithm only in one case causes a % increase in the file size that exceeds 5%. Fig. 4 shows the % difference of the average PSNR performance of the encoder for the foreman sequence of various number of frames various QPs at coding rate 15 frames/sec. The % difference in this case in negligible.

The above results are produced by the values presented in Tables 2 and 3. Table 2 shows the size of files (in bits) produced by the encoder for various numbers of frames and various QPs. Table 3 shows the average PSNR performance of the two

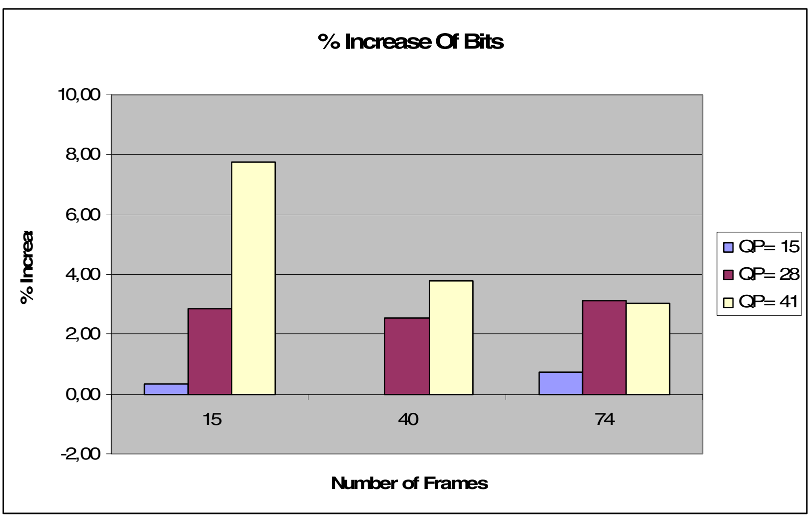

Fig. 3. Percentage of the increase in the file size produced by the encoder with the new algorithm instead of SAD

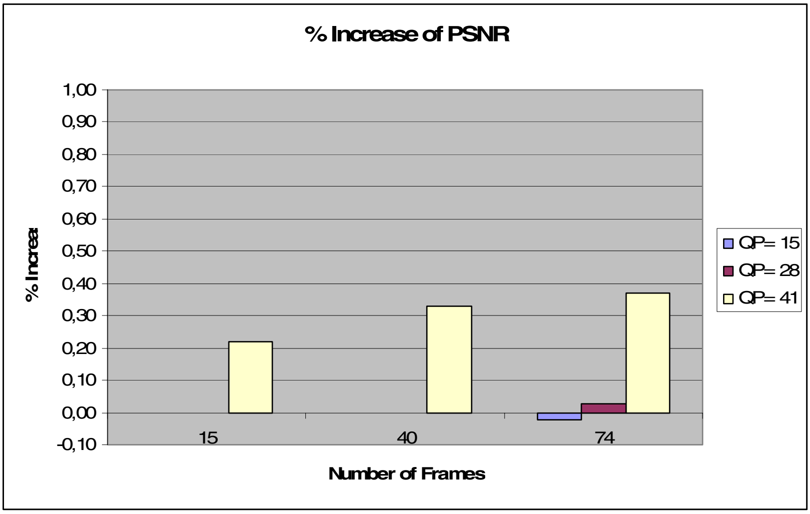

Fig. 4. Percentage of the increase in PSNR produced by the encoder with the new algorithm instead of SAD

Table 2. File size(in bits) comparison

S =SAD P= Proposed		Bits/File {Number of frames=15~74] 15	40	74
QP= 15	S	587160	1520848	2955568
	P	589160	1520752	2977432
QP= 28	S	127208	317464	598776
	P	130960	325816	617960
QP= 41	S	28128	69640	133328
	P	30496	72384	137472

Table 3. PSNR Comparison

S =SAD P= Proposed		PSNR (db) {Number of frames=15~74] 15	40	74
QP= 15	S	45.83	45.83	45.75
	P	45.83	45.83	45.74
QP= 28	S	36.12	36.10	35.65
	P	36.12	36.10	35.66
QP= 41	S	27.32	27.38	26.97
	P	27.38	27.47	27.07

compared algorithms for various numbers of frames and various QPs. From the results presented it can be approved that the proposed algorithm achieves nearly the same efficiency as that of SAD, with a significant reduction in the complexity of the encoder.

5 Conclusions

To conclude, the proposed algorithm by replacing the addition presented in the SAD algorithm with simple comparison, can be implemented in a hardware architecture which, by reducing the complexity, can achieve quite smaller time of execution and a grate reduction in power consumption. The algorithm was integrated with the H.264 JM9.3 codec for the performance evaluation, and the results show that the performance of the proposed algorithm is almost the same with that of SAD.

References

1. ITU-T Rec. H.264 / ISO/IEC 11496-10, "Advanced Video Coding", Final Committee Draft, Document **JVT-J010d1**, December 2003
2. Thomas Wiegand, Gary J. Sullivan, Gisle Bjontegaard, and Ajay Luthra, "Overview of the H.264/AVC video coding standard, " *IEEE Transactions on Circuits, System and Video Technology*, vol. 7, no. 1-19, July 2003.
3. Iain E G Richardson, "H.264 and MPEG-4 Video Compression – Video Coding for Next-generation Multimedia", John Wiley & Sons, 2003
4. JVT Test Model Ad Hoc Group, .Evaluation sheet for motion estimation,. in *ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6*, Draft version 4, February 2003.