Multiple Modes Intra-Prediction in Intra Coding

Peng Zhang, Debin Zhao, Siwei Ma, Yan Lu, Wen Gao
Institute of Computing Technology, Chinese Academy of Science
E-mail: {zhangpeng, dbzhao, swma, ylu, wgao}@jdl.ac.cn

Abstract

Intra-prediction is a widely used technique in intra coding. In H.264 nine directional prediction modes are used and in AVS six modes are used. This paper proposes a multiple mode intra prediction scheme to achieve a better balance between prediction precision and side information cost. Three kinds of additional modes are introduced here to increase prediction precision from different aspects. A significant performance gain can be achieved using this multiple modes intra-prediction while with little complexity increase.

1. Introduction

TV broadcasting and home entertainment has been revolutionized by the advance in video coding technology. Two major groups working on the standardization of this technology, the ITU-T Video Coding Experts Group (VCEG) and ISO Motion Picture Experts Group (MPEG), cooperated to form the Joint Video Team (JVT), and developed up-to-date the most effective video compression standard H.264/MPEG4-AVC (Part10)[1].

In contrast with previous standards, H.264 provides more macroblock coding modes and the block size can vary from 16x16 to 4x4. This change brings us many benefits in coding performance, such as 4x4 integer transform, more precise inter-prediction. On the other hand, due to the weakened decorrelation and energy concentration ability of the small size transform, the variance of transform coefficients increase. It costs more bits to encode the residual in entropy coding. Intra-prediction is introduced to solve this problem. After prediction, the variance of the transform input, which is also named the prediction error, becomes smaller, leading to smaller transform coefficients.

In the finalized H.264 standard, nine prediction modes are used to process intra-prediction. They consist of 8 directional modes using the neighboring reconstructed pixels from 8 directions as the predictors and a DC mode using the average of those pixels as the predictors. [1][4][5][6][7][8][9]

In previous standards, only frequency domain prediction is used for intra coding, such as DC prediction in MPEG2/H.261, and additional AC prediction in H.263/MPEG4. Prior prediction in spatial domain is justified to be suitable for H.264's small size integer transform. Hence spatial prediction becomes a promising intra prediction technique, especially cooperating with integer transform.

China Audio-Video coding Standard (AVS)[2] is finalized recently. Compared with H.264, AVS target to high definition video coding, and an 8x8 integer transform is adopted in AVS. Spatial prediction is also proved to be efficient for this larger size integer transform. But in comparison with H.264, 8x8 integer transform has its own characteristics, and this multiple mode scheme especially agrees to the characteristics. Firstly, as a penalty of the computation facility caused by small integer transform coefficients, decorrelation ability of the transform is weaken and not sufficient to abandon spatial prediction, although larger size transform can get better decorrelation effect. Secondly, more side information is needed in order to reduce the prediction error. Because larger block size leads to larger spatial distance between predictors and predictees, and the corresponding smaller correlation between them decreases the prediction precision.. Thirdly, in view of block amount decreasing, several bits are saved in encoding intra prediction mode for each block, which gives us a chance to employ more prediction modes.

At the same performance, fewer bits are used to encode the prediction error due to more precise prediction with more modes; on the other hand, more bits are assigned to encode the prediction modes. Finding an optimal number of modes and properly selecting the modes can get a optimal balance between performance gain by more precise prediction and performance loss by encoding more prediction modes at a certain rate. The purpose of this paper is to find out relatively optimal modes for AVS intra predictions. Three kinds of modes are introduced in Section 2, and then simulations and results are provided in Section 3, followed by a brief conclusion in Section 4.

0-7803-8603-5/04/$20.00

2. Multiple Modes

Three kinds of prediction modes are discussed in following parts.

2.1. Multiple directions

Since 2-D pictures have an inherent characteristic of spatial correlation, directional prediction is an efficient and widely used method to exploit this kind of correlation. For a majority of regions in most pictures, the texture changes continuously within a certain direction. A best prediction direction minimizing rate-distortion cost is selected for each encoding blocks, .

To demonstrate the relationship between the increase of performance gain and that of directions applied, a uniform directional prediction scheme is introduced here. As an empirical fact, the transform process following the prediction process desires its inputs, to some extent, as flat as possible, in order to reduce high-frequency components. But the long distance prediction of H.264, leads to undesired weaken correlation of neighboring prediction pixels, especially in case of directions except vertical and horizontal and diagonal ones. Demonstrated in figure1, two neighboring pixels may be derived from two reference pixels at a distance of more than one pixel.

To solve the problem stated above, an 8-neighbourhood directional prediction scheme introduced here. We use available pixels just in the 8-neighbourhood of the predictee to interpolate the directional prediction values (Figure 2). The available pixels for interpolation consist of reconstruction pixels from neighboring blocks and predicted pixels of the same block, which are presented by solid circle points in Figure 2 as A, B, C, D, E. Then big diamond points are interpolated by nearest two solid circle points a, b, c, d. Further more, small diamond points are interpolated by big diamond points $\alpha\beta\lambda\eta$. Using A, B, C, D, E, we can conduct a four modes prediction, with C, E in a same direction; adding a, b, c, d, an eight modes prediction; and further more $\alpha\beta\lambda\eta$ and accordingly horizontal interpolation points (not shown in Figure2), performing a 16 mode scheme; and so on

If the direction is between dialog right and vertical $Pred(x,y) = (1-\lambda) \times Pred(x-1, y-1) + \lambda \times Pred(x, y-1)$ and if the direction is between dialog left and vertical $Pred(x,y) = (1-\lambda) \times Pred(x, y-1) + \lambda \times Pred(x+1, y-1)$. λ represents the relative distance from two reference pixels. The similar formula can be derived in the case of horizontal directions.

In the scheme, we can get as many directional predictions as we expect, by just increasing the interpolation points. In addition, codec design will benefit much from such a uniform structure.

In terms of complexity, we can roughly estimate it by the number of pixels to be interpolated:

$$(16+8)/2 \times dir/8 \times 2 = 3dir$$

where dir is the direction number, hence the number of reference pixel gaps is $(16+8)/2$ and then number of interpolation at each gap is $dir/8$, and horizontal and vertical directions should be calculated separately. Complexity can be reduced by hierarchical mode search or EIP in [3].

2.2. Multiple resolutions

In the multiple directional spatial prediction scheme described above, two main steps are:

1. get reference pixels and filtering them.
2. get prediction by interpolation pixel by pixel.

For all directions, a uniform filter is placed before the prediction, with coefficients [1, 2, 1]. Reference pixels are smoothed after the filtering, hence the predicted pixel values are also smoothed, which leads to a smooth prediction error and an accordingly less transformed coefficients. Especially, the filtering is very efficient in case of large block size.

As to another aspect, different pictures or even different regions of a certain picture have different extent of smoothness, which gives us a chance to add more modes to adapt these different situations with a filter of different spatial resolution and filter strength.

To simplify the simulation, we use the convolution of multiple homogeneous filters to build a low resolution filter, with more convolutions corresponding lower resolution. For example, in the case of a picture with sharp texture, no filter is applied to the predictors; and in the case of a smooth picture, a filter with coefficients [1, 2, 1] is applied; and in another case of a smoother picture, a filter with coefficients [1, 4, 6, 4, 1] (convolution of two [1, 2, 1]) is applied; and so on. A filter with 8 convolutions is similar with a 4 steps average window, and 16 convolutions for 8 steps:

$$RP_{resol+1}(n) = RP_{resol}(n) * Filter(n)$$

$$Filter(n) = [1,2,1]/4$$

Where $RP_{resol}(n)$ are the reference pixels at the resolution $resol$, $Filter(n)$ is the filter coefficient array. Through an iterative process, reference pixels at each resolution are derived.

With the increase of the number of convolutions applied to the filter, the smoothness of predictors increases. Thus the DC prediction mode in H.264 can

be simulated by filtering with a very low resolution, and long flat coefficients.

2.3. Multiple fashions

In addition to multiple directions and multiple resolutions, more fashions can be applied to predict the pixels in one block using neighboring reconstructed pixels.

A multiple lines fashion takes the weighted average of pixels in two or more lines of neighboring blocks as the reference pixels. More reference pixels introduced in this fashion mean a higher level linear predictor is applied, which leads to a theoretically better prediction precise. For example (Figure 3a):

$$Pred(x,y) = (RP(x,-1) + RP(x,-2)) >> 1$$

where $RP(x,-1)$ and $RP(x,-2)$ are the two reference pixel lines that is shown as solid in Figure 2a.

A split fashion is used for regions with a significant edge through them. We split the block into two parts in the middle by vertical or horizontal direction, for each part a separate prediction direction is used in order to deriving reference pixels from a nearer neighboring block, instead of a uniform direction within one block. For example (Figure 3b):

if $y < 4$ $Pred(x,y) = RP(x,-1)$

else $Pred(x,y) = (RP(x,-1) + RP(-1,y)) >> 1$

where horizontal pixels $RP(-1,y)$ are used in vertical direction prediction.

Another interpolation fashion predicts a pixel in the block with the linear interpolation from two reference pixels of neighboring blocks. In case of regions with texture changed gradually, such as fading, this fashion is much more efficient than other fashions. For example (Figure 3c):

$$Pred(x,y) = (1-\lambda) \times Pred^{*}(x,y) + \lambda \times RP(-1,7)$$

where λ is a distance factor determined by coordinate y, $Pred^{*}(x,y)$ is the according prediction value by normal fashion of the same direction, and $RP(-1,7)$ is the last pixel in the column left to the block, and red lines in the figure present the prediction direction.

More fashions can be applied, but we just use these four fashions to give a comparison to a single fashion.

3. Simulation Result and Discussions

To demonstrate the characteristic of multiple directions, some simplifications are made from H.264 scheme, such as removing the prediction of intra modes, removing the original DC mode, restricting direction numbers as integer power of 2, and a fix-length encoding scheme for intra modes. 8 sequences including cif and qcif format are tested, and the best (foreman.qcif), the worst (container.qcif) and average performance gain are listed in Table1. These results show that a significant performance gain (over 0.2dB comparing with 8 directions as H.264) is achieved by increasing prediction directions. Multiple directions prediction with more than 64 directions makes no sense considering the tradeoff of complexity and performance.

Multiple resolutions prediction is simulated, with the $resol$ ranging from 1 to 16 pixels, accordingly a similar filter tap ranging from 1 to 8 pixels. Also 8 sequences are tested and the best (container.qcif) and the worst (tempete.qcif) and the average is listed in Table2. Multiple fashions prediction is simulated with four fashions to give a comparison to a single fashion. See Table3.

Separate techniques of adding modes can be integrated into an optimized multiple mode intra-prediction scheme. According to the experimental results, we design a 32 directions, 2 resolutions, and 2 fashion scheme and use it onto AVS3.0, and comparison are shown in Figure3. It can be seen that an average PSNR gain of 0.6dB is achieved.

4. Conclusion

In this paper, we propose a multiple directional intra-prediction scheme. Three kinds of additional prediction modes are introduced with analysis and experiments. Simulation results show that a significant performance gain can be achieved by adding multiple intra-prediction. The balance between prediction error decrease and side information increase caused by multiple mode prediction can be achieve by optimizing the mode selection.

5. Acknowledgement

The work has been supported by National Science Foundation of China(60333020), National Fundamental Research and Development Program (973) of China (2001CCA03300).

6. References

1. “Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec. H.264/ISO/IEC 14496-10 AVC)” in Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVTG050, 2003.
2. Audio Video Coding Standard Workgroup of China(AVS), Video Coding Standard FCD1.0, Nov. 2003.

3. Bojun Meng, “Efficient Intra-Prediction Algorithm in H.264”, in Proc. IEEE International Conference on Image Processing, 2003.
4. Q15-F-11.doc, Telenor Satellite Services, Oct 1998，JVT proposal.
5. Q15-F-24.doc, Nokia, Nov 1998, JVT proposal.
6. Telenor_intra.doc, Telenor Satellite Services, Feb 2000, JVT proposal.
7. VCEG L09.doc, Communications Research Centre, Jan 2001, JVT proposal.
8. VCEG N54.doc, RealNetworks, Inc. Sep 2002, JVT proposal.
9. JVT-B080.doc, RealNetworks, Feb 2002, JVT proposal.

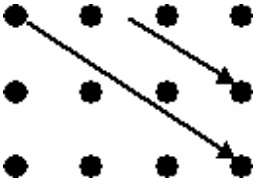

Figure 1 Long distance prediction

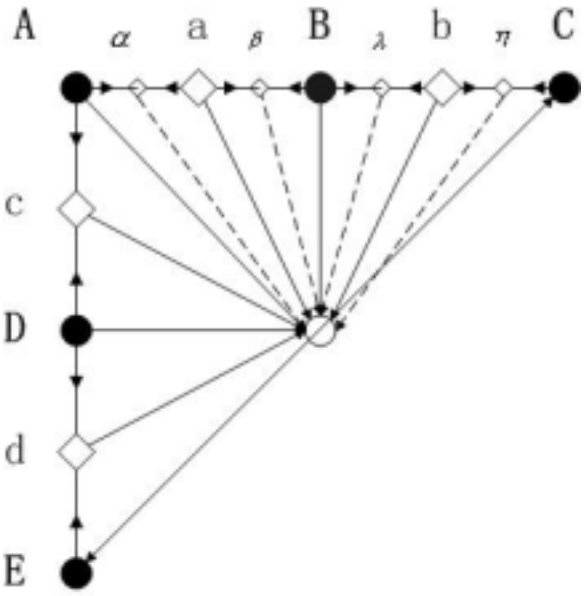

Figure 2 8-neighbourhood prediction

directions		4	8	16	32	64	128
PSNR gain (dB)	best	-0.38	0.00	0.16	0.26	0.30	0.31
	worst	-0.19	0.00	0.07	0.10	0.11	0.10
	avg	-0.25	0.00	0.13	0.21	0.25	0.27

Table 1 multiple direction prediction simulation
Performance gain over 8 directions

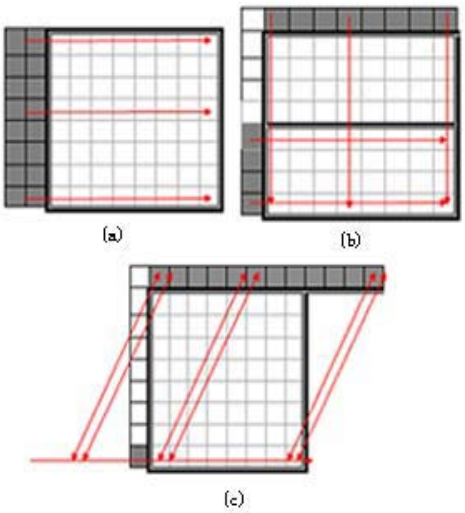

Figure 3 Multiple fashions

resolutions		1	2	4	8	16
PSNR gain (dB)	best	0.29	0.43	0.46	0.43	0.37
	worst	-0.11	0.09	0.19	0.22	0.21
	avg	0.03	0.22	0.29	0.29	0.26

Table 2 multiple resolution prediction simulation
Performance gain over original filter.

		AVS3.0		Multiple Mode		
	QP	PSNR(dB)	Rate(kbps)	PSNR(dB)	Rate(kbps)	APG(dB)
container.qcif	26	43.11	1706.7	43.38	1645.46	0.584641
	35	37.88	971.41	38.13	940.52	
foreman.qcif	26	43.04	1654.21	43.39	1580.35	0.741566
	35	37.68	900.23	38.01	860.66	
news.qcif	26	43.6	1736.09	43.9	1665.26	0.664581
	35	38.17	1014.36	38.46	983.98	
tempete.qcif	26	42.58	2606.75	42.98	2561.01	0.555164
	35	36.61	1546.71	36.95	1525.89	
bus.cif	26	42.68	8150.58	43.07	7939.86	0.614228
	35	36.97	4763.48	37.3	4663.58	
flower.cif	26	43.39	10636.25	43.8	10489.26	0.520483
	35	37.27	6769.9	37.68	6754.66	
foreman.cif	26	43.14	5699.8	43.45	5532.3	0.496458
	35	38.04	2919.23	38.31	2854.53	
mobile.cif	26	42.15	13261.19	42.56	13006.32	0.596466
	35	35.91	8317.22	36.24	8199.46	
average						0.596698

Table 3 an integrated scheme vs. AVS3.0
APG(Average PSNR Gain)