ABSTRACT

H.264/AVC employs intra prediction to reduce spatial redundancy between neighboring blocks. Different directional prediction modes are used to cater diversified video content. Although it achieves quite high coding efficiency, it is desirable to analyze its drawbacks in the existing video coding standard, since it allows us to design better ones. Basically, even after intra prediction, the residue still contains a lot of edge or texture information. Unfortunately, these high frequency components consume a large quantity of bits and the distortion is usually quite high. Based on this drawback, an Edge-based Adaptive Directional Intra Prediction is proposed (EADIP) to reduce the residue energy especially for the edge region. In particular, we establish an edge model in EADIP, which is quite flexible for natural images. Within the model, the edge splits the macroblock into two regions, each being predicted separately. In implementation, we consider the current trend of mode selection and complexity issues. A mode extension is made on INTRA 16x16 in H.264/AVC. Experimental results show that the proposed algorithm outperforms H.264/AVC. And the proposed mode is more likely to be chosen in low bitrate situations.

Index Terms— H.264/AVC, Intra prediction, Edge

1. INTRODUCTION

In most current video coding standards, a frame is divided into multiple non-overlapping blocks. The state-of-the-art video coding standard, H.264/AVC, utilizes different macroblock partitioning sizes, INTRA 4x4, INTRA 8x8 (FRExt) and INTRA 16x16 in intra frame coding. Intra prediction is performed block-by-block. Traditional prediction directions are predefined in the standard as shown in Fig.1a. The prediction is uni-directional like Fig.1b and Fig.1c. The basic assumption is that the texture of the current block follows the same direction. However, this may not be true in block-based coding for natural video content. At object boundaries, an edge may occur in a block, splitting it into two or more regions or segments. The regions often correspond to different objects. Therefore, the pixel values and texture directions may be quite different in different regions. If traditional intra prediction is used, the prediction accuracy may be low resulting in a residue block with large energy to be encoded. Traditional approaches such as those adopted in H.264/AVC try to approximate the complex model of the macroblock (MB) using a piecewise local smooth model via dividing the blocks into smaller squares. However, it leads to increased overhead and decreased energy compaction. This is a drawback of the traditional uni-directional prediction mode.

Another problem of traditional intra prediction is that the prediction directions are quantized (vertical, horizontal, DC, diagonal down-left, diagonal down-right, vertical right, horizontal down, vertical left, horizontal up). However, for natural video, the content direction could be in any direction, instead of just these 9 directions. So the 9 prediction directions are too restrictive for arbitrary video content.

In recent years, for inter frame coding, some relevant approaches have been proposed to address the block-based partitioning limitation. Geometry-adaptive block partitioning [1] attempts to divide the blocks using an arbitrary line segment influenced by the motion boundaries in the scene. This results in a better partition for blocks with discontinuities in the motion field compared with pre-defined squares or rectangles. A simplified scheme is proposed by R.Ferreira et.al in [2], which improves the motion vector prediction and utilizes the joint probability of the partition to reduce the possible partition candidates. Motion-assisted merging [3] or leaf-merging methods [4] attempt to merge different neighboring blocks or sub-blocks based on the similarity of motion vectors. This
method reduces the motion vector bits. All these inter adaptive partition methods share the common feature: providing better prediction with fewer bits. Previous literature [5] illustrates that 2-D piecewise smooth models lead to a faster decay of distortion with rate compared with traditional quadtree based coding structure. As for intra frame, the edge and texture prediction are still two difficult tasks for traditional intra prediction method, as the residue contains obvious edge or texture information. Therefore, an Edge-based Adaptive Directional Intra Prediction method is proposed to reduce the residue energy especially for the edge region. In this paper, we examine the performance of such an edge model employed in intra frame coding.

The rest of the paper is organized as follows: Section II describes traditional intra prediction drawbacks and the proposed intra prediction model. In Section III, a concrete prediction mode is proposed. Simulation results and further analysis are given in Section V.

2. TRADITIONAL INTRA PREDICTION

Traditional block-based intra prediction enables simple and efficient coding of video content. Variable block sizes (4x4, 8x8 and 16x16) further improves coding efficiency. For macroblocks which contain edges, smaller blocks tend to be chosen since it is difficult to predict such macroblocks using one prediction direction model alone. However, the resulting blocks often do not represent the true shape or edge of the objects. Although it models a coarse approximation of the complex macroblock, it leads to individually code blocks with similar data and unnecessary overhead. In H.264/AVC, the intra macroblock mode decision is made upon the Rate-Distortion (RD) criterion. Basically INTRA 4x4 and INTRA 8x8 are chosen where there are texture or edge areas, while INTRA 16x16 are more likely to be chosen since it is difficult to predict such macroblocks using one prediction direction model alone. However, the resulting blocks often do not represent the true shape or edge of the objects. Although it models a coarse approximation of the complex macroblock, it leads to individually code blocks with similar data and unnecessary overhead. In H.264/AVC, the intra macroblock mode decision is made upon the Rate-Distortion (RD) criterion. Basically INTRA 4x4 and INTRA 8x8 are chosen where there are texture or edge areas, while INTRA 16x16 are more likely to be chosen in relatively smooth areas.

The intrinsic goal of the block-based intra prediction is to find blocks that fit the object shape and signal orientation given the reconstructed boundary pixel values. It is also likely that an edge crosses the block in two or more parts, each having different pixel values and directions. And the two regions are in arbitrary shape rather than simple squares. Therefore, edge-based adaptive directional intra prediction is a reasonable extension of the traditional intra prediction.

Sometimes, a block can contain one or more edges and different content orientations in different partitions. In such cases, traditional intra prediction cannot predict well, leaving large residue especially along the edge. When an edge crosses a block, it can split the block into at least two parts. For simplicity, we assume only one edge crosses a 16x16 MB. And the edge is a straight line shown in Fig.2. The straight line splits the current MB into two regions. Each region corresponds to an object with its own content orientation property. Ideally, the partition is expected to be exactly on the object boundary. However, due to the limitation of optical sensors and illumination effects etc., the partition line dose not exactly lie on the object boundary. Actually, this region is called a transition region, from Region 1 to Region 2. The characteristics of the transition region is that: even though the pixels are very close to each other, the values are quite different. Therefore, this region plays a pivotal role in prediction accuracy. In the following, we are going to elaborate on the Edge-based Adaptive Directional Intra Prediction (EADIP) model, which includes how to determine each region (Partitioned Region 1, 2 or Transition Region) and how to predict each region.

3. EDGE-BASED ADAPTIVE DIRECTIONAL INTRA PREDICTION

In EADIP, we construct a coordinate system with its origin at the center of the current MB. The vertical axis is $y$ and the horizontal axis is $x$. For each line, we use two parameter to describe it: $r$ and $\theta$. The distance $r$ is defined as the perpendicular distance from the origin to the edge, in the range of $[0,7]$ with precision $\Delta_r = 1$. The angle $\theta$ is the angle in the range of $[0,360^\circ])$ between the positive horizontal axis and the perpendicular line of the edge with the precision of $\Delta_\theta = 180^\circ / 8$. Given these two parameters, the edge is fixed and can be expressed in a first order way as follows:

$$ f(x, y) = y\sin\theta + x\cos\theta - r = 0 \quad (1) $$

Consider any position $(x, y)$ in the block. If $f(x, y) = 0$, then the pixel is on the edge. If $f(x, y) < 0$, the pixel is defined to be in Region 1. If $f(x, y) > 0$, the pixel is defined to be in Region 2. According to this, we can distinguish whether each pixel belongs to edge (transition) region, Region 1 or Region 2. However, for real applications, the pixel locations are discrete, which means the vertical position $y$ and horizontal position $x$ are both discrete numbers. And the edge may cut through the pixel anywhere within the pixel. Due to this,
in implementation, we define a threshold as follows:

\[
\text{Pixel}(x, y) = \begin{cases} 
\text{Region1}, f(x, y) < -\text{threshold} \\
\text{TransitionRegion1}, |f(x, y)| < \text{threshold} \\
\text{Region2}, f(x, y) > \text{threshold}
\end{cases}
\]  

(2)

The threshold can be any value, which controls the edge width. In our implementation, we set it to be 0.5. After defining the Transition region, Region 1 and Region 2, prediction is performed for transition area and partitioned regions separately as follows.

4. PARTITIONED REGION PREDICTION

After the partitioning mentioned above, Region 1 and Region 2 are defined based on Equ. (2). Each region corresponds to an object. Each object can have its own orientation. Therefore, we allow two prediction directions in EADIP. For each region, the prediction is performed as follows.

When the prediction direction is \( \phi \), the prediction value for each pixel in that region is obtained by constructing a line through the pixel in direction \( \phi \) to find the intersection points with the reconstructed boundaries. For example, in Fig. 2, for the pixel \((x, y)\) in Region 1, a line is drawn along direction \( \phi \). This line crosses the reconstructed boundaries at two points, \( Q_1 \) and \( Q_2 \). We restrict \( Q_1 \) and \( Q_2 \) to be within a certain distance from the current block (16 neighboring pixels on the left, 1 on the up-left, 16 on the top and 16 on the up-right shown in the white color). If it is out of this range, it is considered unavailable. Sometimes, only one point exists depending on the direction.

If two crossing points are available, the average of them is used as prediction value for pixel \((x, y)\). If only one point is available, this point is used as prediction value. If neither point is available, the average of the nearest available boundary points are used as prediction value.

Within one MB, there are two regions with their prediction directions \( \phi_1 \) and \( \phi_2 \) respectively. The allowable range of the prediction direction \( \phi_1 \) and \( \phi_2 \) is \([0, 180^\circ]\). The precision \( \Delta_\phi \) can be controlled according to the users’ need, QP and bitrate. In the current implementation, the precision \( \Delta_\phi \) is \(180^\circ/8\) and thus the orientation range is \([0, 7\Delta_\phi]\). Actually, after careful observation from the video sequences, most edge angles can be categorized by using the precision of \(180^\circ/8\).

5. TRANSITION REGION PREDICTION

For the sake of optical sensors and illumination effects, for the pixels located on the edge, such as the pixel in Fig.3, the pixel value can be calculated using a linear weight combination. Within this pixel, let’s denote the area percentage which belongs to Region 1 as \( W_1 \) and the area percentage which belongs to Region 2 as \( W_2 \). Here, \( W_1 + W_2 = 1 \). When we classify this pixel into Region 1 using the partitioned region prediction method above, we can get a prediction value \( P_1 \). Similarly, when we classify this pixel into Region 2, we can get another prediction value \( P_2 \). And the final prediction value is achieved by the following linear combination.

\[
P(x, y) = W_1 \times P_1 + W_2 \times P_2
\]  

(3)

Basically, if the percentage is larger, a higher weight is given to that predictor. For computational complexity concern, the weight \( W \) is set to be \(1/3, 2/3\) or \(1/2\). The problem is how to decide which region percentage is larger within this pixel. Actually, it is very straightforward when we substitute the current pixel \((x, y)\) into Equ. (1), we know where the centroid of the current pixel is located (in Region 1, Region 2 or on the edge).

6. SIMULATION RESULTS

In EADIP, four parameters \((r, \theta, \phi_1, \phi_2)\) should be encoded as side information. For the distance \( r \), zero-order Exp-Golomb code is used and the other syntaxes are fixed length coded. One additional flag is used to indicate whether traditional intra prediction is used in INTRA 16x16. The proposed scheme is implemented in INTRA 16x16 in JMKT2.A[6]. BD-Bitrate and BD-PSNR is measured according to BJM add-in proposed by [7], which is a widely used performance evaluation tool. All frames in the testing sequences are coded as intra frame using the following encoder configurations: High profile, INTRA 4x4, 8x8 and 16x16 all on, RDO-Q off, MDDT off, Entropy coding CA VLC, deblocking filter on, QP.Slice=(22,27,32,37). The following parameter precision is used: \( \Delta_r = 1, \Delta_\theta = 180^\circ/8, \Delta_\phi = 180^\circ/8 \) For complexity concern, the weight \( P_1 \) and \( P_2 \) is \(1/3, 2/3\) or \(1/2\). In H.264/AVC, for INTRA 16x16 MB type, SATD is used to choose the best intra prediction mode. In the proposed scheme, SATD is firstly used in both choosing the best EADIP mode (among 8x16x8x8 possible prediction candidates) and the best H.264 16x16 prediction mode (among 4 prediction candidates). Given the two best 16x16 prediction modes, RDO is calculated in Equ.4 to further determine the final intra prediction mode for the current MB using INTRA.
Table 1. Bitrate Reduction in High Profile

<table>
<thead>
<tr>
<th>Size</th>
<th>Sequence</th>
<th>Frames</th>
<th>ΔBitrate</th>
<th>ΔPSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIF</td>
<td>Foreman</td>
<td>150</td>
<td>-2.57</td>
<td>0.128</td>
</tr>
<tr>
<td>720P</td>
<td>Spincalendar</td>
<td>40</td>
<td>-2.13</td>
<td>0.093</td>
</tr>
<tr>
<td>720P</td>
<td>Crew</td>
<td>40</td>
<td>-1.87</td>
<td>0.060</td>
</tr>
<tr>
<td>720P</td>
<td>Raven</td>
<td>40</td>
<td>-0.39</td>
<td>0.007</td>
</tr>
</tbody>
</table>

16x16 MB type.

$$RD_{\text{cost}} = SSD + \lambda \times Rate$$  \hspace{1cm} (4)

where SSD is the sum of square difference between the original block and the reconstructed block. Rate is the coded bits for the current block. And \( \lambda \) is the Lagrangian multiplier which is predefined in H.264/AVC.

In Table I, coding results of the proposed scheme for four video sequences with different resolution are given. Compared with H.264/AVC intra coding, better R-D performance can be achieved. Indeed, for sequences which contain many obvious edges, the proposed modes are more frequently chosen than for sequences which do not contain many obvious edges. Take Foreman for example, the MB percentage of the proposed modes can achieve 15% at QP=32. The mode selection is shown in Fig.5, the proposed mode being highlighted with white color while the others being H.264/AVC traditional modes with black color. It can be observed clearly that around edges, the proposed modes are highly likely to be chosen. From experiments, we found the percentage changes with QP value. As QP increases, the percentage of the proposed mode increases as well. In other words, the proposed mode is more likely to be chosen in low bitrate situations.

7. CONCLUSION

Traditional intra prediction scheme achieves quite high coding efficiency. However, due to the limitation of the traditional intra prediction methods, there is still large residue around edges or texture region, which degrades the RD performance. In order to further exploit the spatial redundancy, an Edge-based Adaptive Directional Intra Prediction is proposed to reduce the residue energy for edge regions. Compared with H.264/AVC, RD improvement can be achieved. However, within the proposed model, how to efficiently code the overhead is still an open issue. Sometimes, the large overhead will eat up the improvement made by enhancing the prediction accuracy.

8. REFERENCES


