ADVANCED RESIDUAL PREDICTION IN 3D-HEVC

Xiang Li, Li Zhang and Ying Chen

Qualcomm Technology Inc.
San Diego, CA, USA
{lixiang, lizhang, cheny}@qti.qualcomm.com

ABSTRACT

Inter-view residual prediction (IVRP) is employed to efficiently code non-base texture views by exploiting the correlation of residues between two views in 3D video coding extension of HEVC (3D-HEVC). To further improve the performance of texture coding, advanced residual prediction (ARP) is proposed in this paper. In IVRP, residues in a non-base view are predicted from decoded residues in base view. In contrast, ARP makes the residual predictor for the non-base view block based on newly generated base-view residues by applying motion vector at the non-base view to the base view. Moreover, an adaptive weighting factor is applied to reduce prediction error. Comprehensive simulations show that up to 6.0% luma BD-rate reduction was obtained over IVRP in the reference software of 3D-HEVC.

Index Terms— Inter-view residual prediction, 3D-HEVC, multiview video coding

1. INTRODUCTION

Stereoscopic video is emerging in various applications, including digital cinema, TV broadcasting and consumer electronics. Advanced applications such as auto-stereoscopic 3D TV and free-viewpoint video [1] are becoming possible with better 3D acquisition and display technologies. To better support various stereoscopic or 3D video applications, Motion Picture Experts Group (MPEG) has started the 3D video (3DV) standardization efforts which are currently continued in Joint Collaborative Team on 3D Video coding (JCT-3V) based on the Multi-view Video plus Depth (MVD) data format [2][3]. JCT-3V develops 3DV standards based on H.264/AVC and High Efficiency Video Coding (HEVC) [4]. 3D-HEVC is one of the standard tracks, and targets at developing coding tools for both texture views and depth views of the MVD video data. In 3D-HEVC, texture views are coded independently from depth views. This paper only focuses on the coding of texture, namely multi-view video coding.

In 3D-HEVC, one view is compatible to HEVC in a way that the extracted base view components (e.g., with a layer identifier equal to 0) are decodable by an HEVC decoder to reconstruct a 2D video sequence. Other views, namely non-base views, are coded by a hybrid of HEVC coding tools and inter-view prediction methods. 3D HEVC includes inter-view prediction methods such as disparity motion compensation, inter-view motion prediction and inter-view residual prediction (IVRP) [5]. The latter two both require a disparity vector to locate a corresponding reference block in a base view.

Inter-layer residual prediction was proposed to improve the coding efficiency of scalable video coding (SVC) extension of H.264/AVC [6]. In the context of multi-layer SVC, the same scene is coded several times in different layer representations. Motion compensated residues of the same object in different layers are assumed to be highly correlated. Accordingly, residual block of base layer can be used to further reduce the energy of the co-located block (after taking into consideration of possible up-sampling) of enhancement layer, so that the remaining residues of the enhancement layer block can be more efficiently coded.

In SVC, blocks of the same object in two layers are co-located. However, in multi-view video coding, disparity vector is required to align the coded blocks of the same object. In 3D-HEVC, the disparity vector can be derived based on the motion information of spatial and temporal neighboring blocks [7] so that inter-view residual prediction
Figure 2 Prediction structure of proposed ARP.

becomes possible and more accurate than e.g., assuming zero disparity.

Figure 1 illustrates the basic idea of IVRP in 3D-HEVC. Without loss of generality, uni-directional prediction is assumed for convenience. Given a block $D_c$ in the current view at time $T_j$, let $B_c$, $D_j$ represent its reference block in the base view and its temporal prediction from its reference picture in the same view at time $T_i$, respectively. The prediction $D_c^p$ for block $D_c$ is calculated by as follows:

$$D_c^p = D_c + r_c$$  \hspace{1cm} (1)

where $r_c$ represents reconstructed residues in block $B_c$, which is located by disparity vector derived based on spatial and temporal neighboring blocks [7]. If the disparity vector points to a sub-sample location, the samples of residual reference block are obtained by interpolating the residual samples of the base view with a bi-linear filter [5]. With the prediction in (1), the reconstruction $\hat{D}_c$ of IVRP block $D_c$ is

$$\hat{D}_c = D_c^p + r_c = D_c + r_{B_c} + r_c$$  \hspace{1cm} (2)

where $r_c$ denotes the reconstructed residues in block $D_c$.

Although the above inter-view residual prediction further improves the overall quality of prediction, it has two drawbacks. First, it uses residues of block $B_c$ in base-view to predict those in non-base view regardless how the residues are generated. When the temporal reference of $B_c$ belongs to an object other than the one temporally referenced by block $D_c$, the correlation between residues of $B_c$ and those of $D_c$ will be quite low. Consequently, the IVRP performance will be sub-optimal. Second, even when the residues in block $B_c$ and $D_c$ belong to the same object, the prediction may still be inefficient due to the disparity and quantization differences between views. To improve the efficiency of IVRP, advanced residual prediction (ARP) [8] is proposed in this paper. Simulations under 3D-HEVC common test conditions [9] show that up to 6.0% luma BD-rate reduction was achieved by ARP over IVRP for non-base views.

The remainder of the paper is organized as follows. First, the proposed method is introduced in section 2. Subsequently, simulation results and discussions are provided in section 3. Finally, the paper is concluded in section 4.

2. ADVANCED RESIDUAL PREDICTION FOR 3D-HEVC

In this section, the idea of advanced residual prediction (ARP) is firstly introduced in section 2.1. Then two key parts of ARP, namely derivation of residual predictor in base view and determination of weighting factor, are discussed in section 2.2 and 2.3, respectively. Finally, the implementation summary is provided in section 2.4.

2.1 Advanced Residual Prediction

ARP targets at improving residual prediction for non-base views.

Figure 2 illustrates the prediction structure of ARP in multiview video coding. Still, uni-directional prediction is assumed in the following discussions for simplicity. The proposed method can be easily extended to the case of bi-directional prediction.

As previously mentioned in Section 1, the performance of IVRP will drop when residues in $B_c$ belong to an object other than the one of block $D_c$. To avoid such cases, a reference block $B_c$ in the base view at time $T_i$ is located based on the relative positions among $D_c$, $D_j$, and $B_c$, such that $B_c$ is a projection of $D_c$ in the base view. Considering $B_c$ is actually the projection of $D_c$ in the base view and $D_j$ serves as the temporal reference of $D_c$ in the dependent view, the difference between $B_c$ and $B_j$ should be highly correlated with that between $D_c$ and $D_j$. Consequently, residues $r_{B_c}$ in base view obtained by (3) can be used to predict those in non-base view.

$$r_{B_c} = B_c - B_j$$  \hspace{1cm} (3)

On the other hand, due to the disparity between views, the same object may show different behavior in different views, such as translational motion in one view may become zooming in/out in another view. Therefore, it is sometimes not efficient to directly use residues $r_{B_c}$ obtained by (3) to predict residues in the non-base view. Ideally, disparity based warping operation on $r_{B_c}$ shall be applied before the prediction to accommodate the disparity between views. However, the computational complexity of such operations is too high for practical codec. Moreover, quantization difference between base and non-base views also leads to less prediction accuracy of (3). To partially solve the problem, a simple yet efficient approach is proposed by introducing a weighting factor $w$ on top of the residues so that the proposed prediction $D_c^{p'}$ of $D_c$ is

$$D_c^{p'} = D_c + w \cdot (B_c - B_j).$$  \hspace{1cm} (4)

Accordingly, the reconstruction of block $D_c$ is

$$\hat{D}_c = D_c^{p'} + r_c = D_c + w \cdot (B_c - B_j) + r_c.$$  \hspace{1cm} (5)
where \( r_c \) denotes the un-quantized residues in block \( D_c \).

The proposed ARP method is formulated by (4) and (5). Compared to IVRP in (1) and (2), there are two key differences. First, instead of directly using residues in block \( B_r \), residual predictor is derived as the difference between \( B_r \) and \( B_r \) (the projection of \( D_r \)’s temporal reference in the base view). Since the residue predictor and those to be predicted in \( D_c \) belong to the same object, the prediction performance is improved. Second, weighting factor \( w \) is introduced to further compensate disparity and quantization difference in different views.

2.2 Derivation of Residual Predictor in Base View

The residue predictor is derived on the fly during encoding/decoding. The key part of the derivation is to get the reference block \( B_r \) in the base view, for which several approaches may be applied in practice. In the current implementation, \( B_r \) is achieved based on the motion vectors of the current Coding Unit (CU). As shown in Figure 2, block \( B_r \) is first located from \( D_r \) with disparity vector, as in IVRP. Then block \( B_r \) is obtained by applying \( D_r \)’s motion vector \( V_{rb} \) to \( B_r \), Additional motion compensation in the base view picture therefore is invoked for the reference block. Afterwards, residual predictor \( r_{bc} \) is calculated by (3).

2.3 Determination of Weighting Factor

The indication of whether ARP is enabled for the current CU and the exact weighting factor is independent to the residue predictor generation.

Theoretically, weighting factor \( w \) can be calculated by minimizing the difference between the original signal \( I_c \) of the block \( D_c \) and the prediction \( D^p_c \) in (4), namely

\[
w = \arg\min_w [||I_c - D^p_c||].
\]

(6)

where the operator \( || \cdot || \) represents \( L^2 \)-norm. However, minimizing distortion error cannot guarantee the optimal performance in terms of rate-distortion. Moreover, arbitrary \( w \) value may be obtained from (6), which is expensive to code in bitstream.

As a tradeoff, \( w \) is selected from a predefined candidate list. Three weighting factors, i.e., 0, 0.5, and 1, are employed. The one leading to the minimal rate-distortion cost for the current CU is selected as the final weighting factor. Then corresponding weighting index (from 0 to 2) is transmitted in bitstream.

2.4 Implementation Summary

The proposed ARP algorithm was implemented on top of the reference software of 3D-HEVC, i.e., 3D-HTM version 4.0. The ARP weighting index is signaled with context-adaptive binary arithmetic coding (CABAC) for each inter coded CU in non-base texture views. Please note that (4) and (5) apply to both luma and chroma components. When a CU contains more than one Prediction Unit (PU), all PUs share the ARP weighting factor signaled for this CU. Similarly, luma and chroma components in a CU share the same weighting factor.

At encoder side, Algorithm 1 applies for ARP mode selection. At decoder side, (5) is employed for the reconstruction. In the case of bi-prediction, ARP applies to each direction and the average of the two predictions serves as the final prediction of a PU.

3. SIMULATION RESULTS AND DISCUSSIONS

In order to verify the performance of the proposed ARP, it was implemented on top of 3D-HEVC reference software 3D-HTM 4.0 [5]. Simulations under the common test conditions [9] for 3D-HEVC were conducted except that no depth views were coded since both IVRP and ARP only apply to texture views. For performance comparison, reference software 3D-HTM 4.0 with IVRP enabled was taken as the anchor. Note that the parsing problem of IVRP was avoided according to [10]. Since base view was always identically coded, it was excluded from performance comparison.

Table 1 summarizes the simulation results, where “View 1” and “View 2” denote two non-base texture views. The coding performance is measured in BD-rate [11] reduction for luma component, as requested in [9]. Note that a negative number indicates coding gain. More results including chroma performance may be found in [12]. From the table, it can be easily observed that ARP outperforms IVRP for all test sequences. On average, ARP, as a new coding technique, provides 3.5% luma BD-rate reduction for each non-base view. Up to 6.0% luma BD-rate reduction was obtained for the second non-base texture view of sequence Kendo.

Figure 3 shows the rate distortion curves of the luma component of the Kendo sequence at different bit-rates. It can be noticed that ARP is much more efficient than IVRP in the reference software. Particularly, higher coding gain is observed at high rate. Actually, residues consume relatively...
more bits at high rate. In this case, residual prediction becomes more important. With ARP, residues are more efficiently predicted so that better performance is achieved.

The computational complexity of ARP is higher than that of IVRP. When compared to IVRP, encoding and decoding time was increased by around 20% and 6%, respectively. At encoder side, the computationally intensive part of ARP is the module of ARP mode selection where proper weighting mode needs to be decided based on rate-distortion cost. To improve encoding speed, fast algorithm such as early termination may be employed. At decoder side, the most time consuming part of ARP is the calculation of the weighted difference between base view block $B_c$ and block $B_r$. Since such block operations are quite regular, parallel computing will help much in practice.

4. CONCLUSION AND FUTURE WORK

In this paper, advanced residual prediction (ARP) is proposed to efficiently code texture views in 3D-HEVC. With the consideration of both temporal motion and disparity information, residual predictor is derived more accurately. Moreover, adaptive weighting factor is introduced to further accommodate disparity and quantization difference between views. Simulations were conducted under the JCT-3V common test conditions. Compared to the inter-view residual prediction design in 3D-HEVC, up to 6.0% luma BD-rate reduction was obtained for non-base views. Due to the good performance, ARP has been adopted into HEVC based 3D-HEVC in 4th JCT-3V meeting.

For the next step, low-complexity and hardware friendly ARP design is to be studied.

5. REFERENCES


<p>| Table 1 Coding performance of ARP |</p>
<table>
<thead>
<tr>
<th>Resolution</th>
<th>Sequences</th>
<th>View 1</th>
<th>View 2</th>
<th>Aver of two views</th>
</tr>
</thead>
<tbody>
<tr>
<td>1024x768</td>
<td>Balloons</td>
<td>-5.0%</td>
<td>-4.3%</td>
<td>-4.7%</td>
</tr>
<tr>
<td></td>
<td>Kendo</td>
<td>-5.2%</td>
<td>-6.0%</td>
<td>-5.6%</td>
</tr>
<tr>
<td></td>
<td>Newspaper</td>
<td>-3.6%</td>
<td>-3.0%</td>
<td>-3.3%</td>
</tr>
<tr>
<td>1920x1088</td>
<td>GTFly</td>
<td>-1.9%</td>
<td>-1.9%</td>
<td>-1.9%</td>
</tr>
<tr>
<td></td>
<td>PoznanHall</td>
<td>-1.6%</td>
<td>-1.7%</td>
<td>-1.7%</td>
</tr>
<tr>
<td></td>
<td>PoznanStreet</td>
<td>-3.9%</td>
<td>-4.1%</td>
<td>-4.0%</td>
</tr>
<tr>
<td></td>
<td>UndoDancer</td>
<td>-3.3%</td>
<td>-3.3%</td>
<td>-3.3%</td>
</tr>
<tr>
<td>Aver of 1024 sequences</td>
<td>-4.6%</td>
<td>-4.4%</td>
<td>-4.5%</td>
<td></td>
</tr>
<tr>
<td>Aver of 1920 sequences</td>
<td>-2.7%</td>
<td>-2.8%</td>
<td>-2.8%</td>
<td></td>
</tr>
<tr>
<td>Aver of all</td>
<td>-3.5%</td>
<td>-3.5%</td>
<td>-3.5%</td>
<td></td>
</tr>
</tbody>
</table>

Figure 3 Rate-distortion curves of the second non-base view for Kendo