Improved Intra Moding Coding Beyond AV1

Yize Jin, Liang Zhao, Xin Zhao, Shan Liu, Alan Bovik

Tencent America, Palo Alto, CA, USA.

The University of Texas at Austin, Austin, TX, USA

Abstract

In AOMedia Video 1 (AV1), directional intra prediction modes are applied to model local texture patterns that present certain directionality. Each intra prediction direction is represented with a nominal mode index and a delta angle. The delta angle is entropy coded using shared context between luma and chroma, and the context is derived using the associated nominal mode. In this paper, two methods are proposed to further reduce the signaling cost of delta angles: cross-component delta angle coding, and context-adaptive delta angle coding, whereby the cross-component and spatial correlation of the delta angles are explored, respectively. The proposed methods were implemented on top of a recent version of libaom. Experimental results show that the proposed cross-component delta angle coding achieved average 0.4% BD-rate reduction with 4% encoding time saving over all intra configurations. By combining both methods, an average 1.2% BD-rate reduction is achieved.

Introduction

- In video coding standards developed in the past decade, such as High-Efficiency Video Coding (HEVC), Versatile Video Coding (VVC), VP9 and AOMedia Video 1 (AV1), intra prediction has played a vital role for reducing spatial redundancy among neighboring samples within one picture. The typical intra prediction modes (IPMs) in a hybrid coding based video codec can be classified into directional modes and non-directional modes, where the directional modes subsample the continuous space of the directionalities into limited discrete angles, and the non-directional modes assume that the current coding block contains smooth pixel value transitions within the current block without a dominant direction.

- In HEVC, a total of 35 IPMs are defined for intra prediction, including 33 directional prediction modes and 2 non-directional modes, i.e., Planar and DC mode. In VVC, a finer granularity of directional intra prediction modes is adopted. In VP9, the intra prediction applies 10 IPMs, including 8 directional modes and 2 non-directional modes, i.e., DC and true motion (TM) modes. In AV1, the directional modes in VP9 are termed as nominal modes, and one of 7 delta angles with index ranging from -3 to +3 is further signaled for each nominal mode to increase the granularity of the orientations, which defines 56 directional modes overall.

- In this paper we propose a cross-component mode coding method and a context-adaptive mode coding method, which exploits the correlation of delta angles between chroma and luma components, and adjacent luma components, respectively. We show that the performance of the first method achieves 0.4% BD-rate reduction and 4% encoding time saving, and combining both methods achieves 1.2% coding gain and 56% increase in encoding time.

Intra Prediction

- In AV1, signaling of intra mode coding requires at least four syntaxes: the nominal angle of luma (y_mode), the delta angle of luma (angle_delta_y), the nominal angle of chroma (uv_mode), and the delta angle of chroma (angle_delta_uv).

- When searching for the best mode, all the nominal modes are looped by using a predefined and fixed order to search for the best intra prediction mode.

- For the luma component, there is a rough mode decision process to determine which nominal mode can enter full RDO process or not. For the chroma component, all the nominal modes are involved into the full RDO process.

- Since the co-located luma and chroma blocks typically represent the same subject, there is a high correlation between the intra prediction angles of luma and chroma blocks. The percentage of luma and chroma nominal angles being equal is around 95% for non-zero chroma delta angles, and 55% of the luma and chroma delta angles have equal value when luma and chroma is predicted using the same nominal mode. The percentage of the nominal angle of the current luma block being equal to one of the neighboring nominal modes is over 80%.

```
intra_block_mode_info( ) {
   RefFrame[ 0 ] = INTRA_FRAME
   RefFrame[ 1 ] = NONE
   y_mode
   ……
   if ( is_directional_mode( y_mode ) && …) {
      angle_delta_y
   }
   if ( HasChroma ) {
      uv_mode
      ……
      if ( is_directional_mode( uv_mode )  &&  …) {
         angle_delta_uv
      }
   }
}
```

Table 1 : Syntax table for intra mode coding

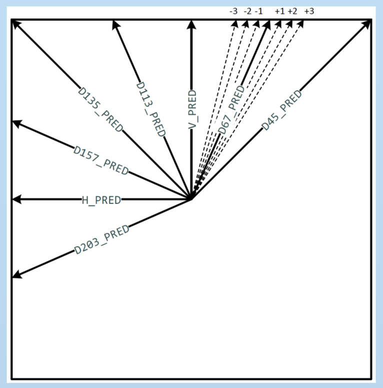

Figure 1 : Illustration of directional intra prediction modes in AV1.

Proposed Method

- Motivated by the observation that chroma delta angles are highly correlated with luma delta angles, instead of using the chroma nominal angles to derive the CDF of the chroma delta angles, we make the CDF of the chroma delta angles dependent on the delta angles of the co-located luma blocks, which further exploits the correlation between the luma and chroma components.

- There are two alternatives to make chroma delta angles dependent on luma delta angles. In the first alternative, namely Method #1, chroma delta angles are only allowed when the luma and chroma nominal angle are same, and the chroma delta angle depends on luma delta angle. In the second alternative, namely Method #2, all chroma delta angles are allowed regardless of whether the luma and chroma nominal angle are same or not. Furthermore, in Method #2, two different CDFs are employed. One is used when luma and chroma nominal angles are the same, whereas the other CDF is used when the luma and chroma nominal angles are not equal.

- The context-adaptive mode coding method is proposed for directional luma modes by conditioning the luma delta angles on its neighboring delta angles. Specifically, two different CDF derivation processes are designed, based on whether the nominal angle of the current block is the same as the nominal angle of one of its neighbors.

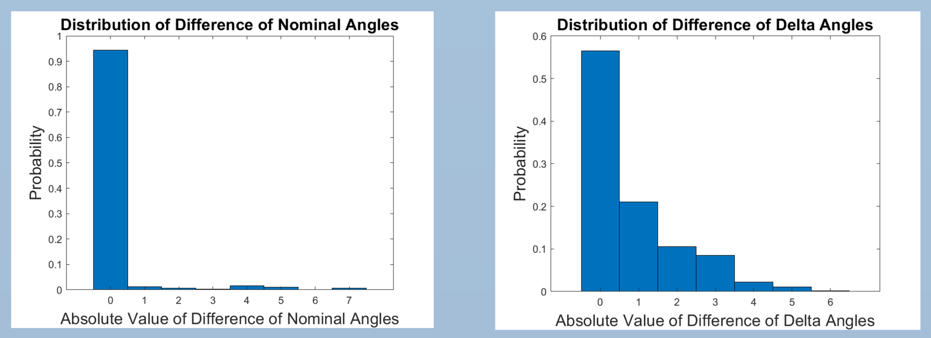

Figure 2 Statistics on the distribution of delta angle for chroma component.

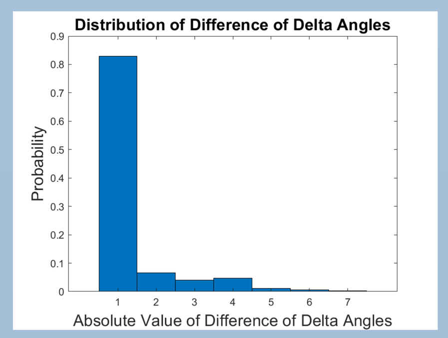

Figure 3 Statistics on the distribution of delta angle between neighboring blocks for luma component.

Experimental Results

Table.1 Summary of coding performance of cross-component mode coding methods

	Method #1					Method #2					Method #2 (High QP)				
	BDR-Y	BDR-U	BDR-V	YUV	ΔT_{Enc}	BDR-Y	BDR-U	BDR-V	YUV	ΔT_{Enc}	BDR-Y	BDR-U	BDR-V	YUV	ΔT_{Enc}
Class A1 (4K)	-1.04%	1.17%	1.24%	-0.30%	96%	-1.02%	0.88%	1.08%	-0.37%	100%	-1.53%	0.42%	1.05%	-0.89%	101%
Class A2 (4K)	-1.05%	1.14%	0.96%	-0.57%	94%	-1.02%	0.83%	0.66%	-0.61%	100%	-1.24%	0.96%	0.75%	-0.81%	100%
Class B (1080P)	-1.03%	2.10%	2.18%	-0.37%	96%	-1.01%	2.00%	1.89%	-0.40%	100%	-1.30%	2.22%	1.93%	-0.63%	100%
Class C (480p)	-0.87%	2.31%	2.40%	-0.26%	97%	-0.84%	1.92%	1.83%	-0.31%	100%	-1.31%	2.83%	2.33%	-0.48%	100%
Class E (720P)	-1.19%	2.02%	1.42%	-0.58%	96%	-1.18%	1.16%	1.05%	-0.70%	100%	-1.54%	1.21%	1.22%	-0.94%	100%
Class D (240p)	-0.63%	1.62%	2.06%	-0.14%	97%	-0.66%	1.07%	1.42%	-0.29%	99%	-1.11%	2.48%	1.81%	-0.45%	100%
Class F (Synthetic)	-0.82%	1.86%	1.41%	-0.30%	97%	-0.78%	1.59%	1.34%	-0.30%	100%	-1.10%	0.46%	0.96%	-0.64%	100%
Average*	-1.03%	1.82%	1.74%	-0.40%	96%	-1.00%	1.46%	1.40%	-0.46%	100%	-1.37%	1.68%	1.56%	-0.72%	100%

Table.2 Summary of coding performance of contex-adaptive coding methods

	Method #3					Method #4				
	BDR-Y	BDR-U	BDR-V	YUV	ΔT_{Enc}	BDR-Y	BDR-U	BDR-V	YUV	ΔT_{Enc}
Class A1 (4K)	-1.10%	-0.62%	-0.64%	-0.94%	151%	-2.05%	0.31%	0.30%	-1.27%	148%
Class A2 (4K)	-1.05%	-0.06%	-0.15%	-0.82%	156%	-1.88%	0.56%	0.34%	-1.35%	152%
Class B (1080P)	-0.99%	0.09%	-0.18%	-0.79%	159%	-1.80%	1.67%	1.72%	-1.07%	157%
Class C (480p)	-0.65%	-0.29%	-0.27%	-0.58%	162%	-1.46%	1.78%	1.93%	-0.83%	162%
Class E (720P)	-1.33%	-0.40%	-0.82%	-1.18%	161%	-2.29%	0.75%	0.21%	-1.71%	160%
Class D (240P)	-0.49%	-0.15%	-0.34%	-0.45%	16a0%	-1.15%	1.27%	1.89%	-0.62%	159%
Class F (Synthetic)	-0.57%	-0.22%	-0.57%	-0.54%	158%	-1.43%	1.21%	1.05%	-0.89%	157%
Average*	-1.00%	-0.22%	-0.38%	-0.84%	158%	-1.86%	1.13%	1.05%	-1.20%	156%

Test condition: The software used to test the algorithms is libaom v2.0.0 [11], with the following hash tag#:

299b96d3836e8fa9b6ee85646175e1dce2a4dd74

All intra coding was used for the test, while cpu-used-0 was used so that the experiments were conducted using the highest quality setup. All the test sequences were defined under JVET common test conditions (CTC) [12]. To get the average BD-rate for the YUV components, the average YUV PSNR was calculated by the following equation, which is used in JVET for tool reporting,

$$PSNR\_YUV = (6 * PSNR\_Y + PSNR\_U + PSNR\_V)/8.$$

The set of QP values are 28, 35, 42, 49, which covers the middle range of QP values.

- **Coding Performance:** The proposed cross-component mode coding methods achieved 0.40% and 0.46% BD-rate savings on average for Method #1 and Method #2, respectively. In addition, the BD-rate reduction is higher for higher 4 QPs (42, 49, 56, 63), which is 0.72%. In addition, Method #1 also reduced encoding complexity by 4%, and Method #2 maintains the same complexity. The results for context-adaptive mode coding, namely Method #3, achieves a coding gain of 0.84%, while increased encoding complexity by 58%. Method #4 combined Method#1 and Method #3, achieving BD-rate savings of 1.20%, while the encoding time increased by 56%.

Conclusion

In this paper, a cross-component mode coding algorithm is proposed to exploit dependencies between luma and chroma delta angles, and a context-adaptive mode coding algorithm is proposed to exploit dependencies between neighboring luma delta angles. By combining these two proposed methods, 1.20% BD-rate reduction is achieved with 156% of encoding time as compared to AV1.