L-shape-based Block Partitioning and Iterative Prediction (L-BPIP) Algorithm for HEVC Intra-Frame Lossless Compression

Qinghao Lin, Min Lin✉, Xiaoyu Weng, Guojie Chen

Abstract—H.26x family of video coding standards designed by ITU (International Telecom Union) is widely adopted in last decades, in which block-based prediction is employed in intra-frame coding. In this paper, we introduce an L-shape-based Block Partitioning and Iterative Prediction (L-BPIP) algorithm in which each block is divided into a series of close neighboring 1-D L-shapes and each L-shape prediction is performed by recursive method with previous neighboring L-shape as reference samples. When predicting each L-shape, similar to traditional block prediction, all the prediction modes are searched to find out the best one with minimum rate-distortion (RD) cost. Because of the L-shapes inherited close neighboring characteristics, L-shape-based iterative prediction can have much more accurate prediction results and much less residual bits which need to be coded and transmitted finally. In addition to smaller prediction residuals, "Mode Residual" is introduced so that most redundancy in increased number of mode information is eliminated due to iterative prediction method. To further maximize the efficiency of our prediction method, this paper also proposes an L-shape-based block partitioning method, which allows a block to be divided into a combination of an L-shaped block and a square block. Statistics shows this block partitioning method increases 15% large blocks compared to traditional method, which can further reduce redundancy in mode information. Experimental results on HEVC reference software HM-16 shows that our proposed merge method is able to achieve an average of 8.87% bit-rate saving, with the maximum of 14.14% bit-rate saving on intra-frame lossless compression.

Index Terms—High efficiency video coding, intra-prediction, lossless coding, iterative prediction, mode residual, L-shape block partitioning, L-BPIP

I. Introduction

HIGH Efficiency Video Coding (HEVC) standard is the most popular video coding standard of the Joint Collaborative Team on Video Coding (JCT-VC) [?]. It can achieve about 50% bit-rate saving in comparison with H.264/AVC [?] for the same quality with a much higher coding complexity [?]. It adopts various coding efficiency enhancement and parallel processing tools. The next generation Versatile Video Coding (VVC) [?] standard developed by Joint Video Exploration Team (JVET) comes with a series of advanced coding tools in an effort to improve the coding efficiency [?], providing exceeding 30% higher compression rate for the same video quality than HEVC. In several generations of the H.26x family of video coding standards designed by International Telecom Union (ITU) including H.265/HEVC and H.266/VVC, block-based prediction is performed in intra-prediction. This process performs angular and planar prediction to the current coding unit (CU) before which a frame is divided into CUs under a quadtree-based block partitioning structure.

In addition to the commonly used lossy video compression, lossless video compression is also very popular in many applications such as automotive vision, web collaboration, remote desktop sharing [?], content creation, post production, and professional applications such as medical imaging [?], and digital preservation in libraries and archives. Thus improving lossless intra-prediction coding efficiency is of great necessity.

To consider the lossless coding mode [?], we simply bypass transform, quantization, and in-loop filters (deblocking filter, sample adaptive offset, and adaptive loop filter) [?].

In video codec standards, intra-prediction takes advantage of the correlation between adjacent pixels in the spatial domain to eliminate spatial redundancy in the image, and inter-prediction uses the correlation between adjacent frames in the temporal domain to eliminate temporal redundancy. Compared to inter-prediction, intra-prediction has great advantages of much simpler hardware structure and much lower power consumption, which is very suitable for the small and portable wearable devices such as light-weighted AR/VR devices. Intra-prediction is also used in the reference frame (I frame) of every Group Of Pictures (GOP) as an important part for inter-prediction. Further more, experimental results on the VVC platform VTM-12.0 shows that for intra-frame lossless compression, VVC provides only 10% gain, with a price of much higher complexity which can be observed to have almost 25 times the encoding time compared to HEVC. That is a critical and detrimental flaw for low power and real-time performance required by wearable AR/VR devices, increasing the significance of improving HEVC intra-frame lossless compression.

For intra-prediction, HEVC block-based prediction can provide accurate prediction because of its flexible block

Q. Lin*, M. Lin✉ are with the Advanced SoC and IoT Technology Lab (ASITLAB), Key Laboratory of Specialty Fiber Optics and Optical Access Networks, Shanghai University, Shanghai 200444, China. (e-mail: shu_rin@shu.edu.cn, mlin99@shu.edu.cn). (Corresponding author: Min Lin).

This work was supported by Ministry of Science and Technology of China, through the National Key Research and Development Program of China, under Grant 2019YFB2204500.

partitioning method, but in the case of predicting large blocks, pixels close to the edge of the block which are far away from the reference samples will inevitably produce larger prediction residuals, resulting in the reduce of the coding efficiency. Secondly, although the traditional block partitioning method can provide an appropriate partitioning result, some of the sub blocks under the same parent block cannot be jointly predicted and coded when there appears some kind of texture mutation in the parent block, even if they choose the same prediction mode and share a common texture. This can cause the problem of increasing redundancy of mode and partitioning information.

Due to these imperfections of traditional algorithm, we design an L-shape-based Block Partitioning and Iterative Prediction (L-BPIP) algorithm in this paper to improve intra-frame lossless compression in HEVC. The main contributions of this work are as follows:

1) Propose a new iterative prediction method.
2) A new L-shape-based block partitioning method is introduced.
3) Based on the two points above, we propose an L-shape-based Block Partitioning and Iterative Prediction (L-BPIP) algorithm.

The iterative prediction method can decrease the minimum distance between the reference samples and the predicted samples, thereby reducing prediction residuals. And by introducing L-shape blocks, we can merge regions with similar textures so that the region containing rich textures can still maintain blocks as large as possible locally, only using small blocks for coding in the highly detailed texture regions. Simultaneously, the form of the L-shape block can better fit the iterative prediction we proposed and further reduce the redundancy in mode information. Through the combination of the two methods, the so-called L-shape-based Block Partitioning and Iterative Prediction (L-BPIP) algorithm is presented for better performance in the intra-frame lossless compression.

The rest of this paper is organized as follows. Section II provides a review of traditional intra-prediction and block partitioning method in HEVC. Section III introduced the proposed L-shape-based Iterative Prediction (L-IP) and L-shape-based Block Partitioning (L-BP) method in detail, as well as the overall framework process of their combined implementation. Section IV introduces the optimization of the scheme and analyzes its effect. Section V provides the experimental results with analysis and discussion. Finally, Section VI concludes this paper.

II. ANALYSIS OF INTRA PREDICTION AND BLOCK PARTITIONING IN HEVC AND RELATED WORK

A. INTRA PREDICTION IN HEVC

Intra-prediction is one of the core parts of video coding. This process uses prediction residuals to replace the pixel data which need to be coded and transmitted so as to perform data compression.

?? shows an example of traditional angular prediction in HEVC. For each certain angular direction, the decoded and reconstructed samples of the nearest neighboring blocks are used as reference samples. Pixels are projected along the angular direction associated with the intra mode to the reference line to get its predicted value. If it doesn't fall on integer grid, the linear interpolations of neighboring reference samples of the intersection are used to generate the predicted value in 1/32 pixel precision:

$$P_{(x,y)} = ((32 - w) \cdot R_{(0,i)} + w \cdot R_{(0,i+1)} + 16) >> 5 \quad (1)$$

where we denote reference sample by $R_{(x,y)}$, and predicted sample by $P_{(x,y)}$. The subscript, (x, y), represents the coordinates of the pixel, which defines the top-left pixel as $(0, 0)$. If some reference samples, like $R_{(N+1,0)}$ $R_{(2N,0)}$ are missing due to prediction order or location of the current CU (such as the picture/slice boundaries), they need to be padded with the nearest available reference sample before the predicting process. w in the function denotes the weight of the reference samples depends on the position of the projection point.

Although in this process, the nearest reference lines along the current block are used, it is the block-based feature that makes the pixels near the edge of the block farther from the reference line which results in the inaccurate predicted value in this area and thus leads to larger prediction residuals. ?? displays the average prediction distortions of all 35 intra modes in 16 × 16 luma prediction units (PUs) [?]. These statistical results are generated by encoding 1st frames of four sequences (Traffic, Kimono, BasketballDrill, and FourPeople) using HM-16.12. Dark area means a small distortion, and bright area represents a high distortion level. It can be clearly observed from the experimental results shown in ?? that, when the distance becomes longer between the pixel and the reference line, the residuals goes up simultaneously. Smaller prediction residuals are located in the region close to the reference line, while the intra prediction is inferior in the farther regions, which is caused by the attenuation of spatial correlation as the prediction distance ascends.

For this flaw of intra-prediction, [?] presented a intra coding method based on sample-by-sample differential pulse code modulation (DPCM), which applies DPCM on residuals after intra prediction and it is used for only lossless coding in H.264/AVC Fidelity Range Extensions (FRExt). And HEVC had modification compared to H.264/AVC FRExt, supporting horizontal and vertical DPCM coding modes for residual signals (with either intra-prediction or inter-prediction) in HEVC Range Extension (RExt) [?], [?]. This kind of pixel-by-pixel DPCM is also named as residual DPCM (RDPCM) since DPCM is applied to the residuals [?]. [?] introduced a sample-based angular intra prediction (SAP) method, which employed the same prediction mode signalling method and the sample interpolation method as the HEVC block-based angular prediction, but uses adjacent neighbors for better intra prediction accuracy and performs prediction sample by sample. Subsequent DPCM-based proposals are SAP-

Fig. 1. Traditional angular prediction in HEVC

Fig. 2. Average prediction distortions of all 35 intra modes in 16 × 16 PUs

HV [?], SAP1 [?], and SAP-E [?]. SAP-HV applies DPCM exclusively in the pure horizontal and vertical directions. SAP1 is similar to SAP but employs a more uniform density of prediction modes in the vertical and horizontal directions. SAP-E has been shown to provide further bit-rate reductions over SAP, SAP-HV and SAP1, as tested on large color biomedical images [?]. [?] analyzed the range of values in residual blocks and apply accordingly a pwm function to map specific residual values to unique lower values and encode appropriate parameters associated with the pwm functions at the encoder, so that the corresponding inverse pwm functions at the decoder can map values back to the same residual values. [?] exploited inter-channel correlation using reconstructed luma to predict chroma linearly with parameters derived from neighboring reconstructed luma and chroma pixels at both encoder and decoder to avoid overhead signalling. Cross-Component Linear Model (CCLM) for chroma intra-prediction is a promising coding tool in Joint Exploration Model (JEM) developed by the Joint Video Exploration Team (JVET). [?] introduced an approach which can classify neighboring luma and chroma samples of the current block into several groups, and derive a particular set of linear model parameters for each group, and a new prediction method synthesizing the angular intra-prediction and the MM-CCLM intra-prediction into a new chroma intra coding mode. [?] made full use of reconstructed pixels to predict later ones in bidirectional or multi-directional manner by coding pixels alternately with interleave prediction. In [?], to reduce residual energy, half pixels are coded via a novel padding technique together with a constrained quantization algorithm, whereas the other half are reconstructed by linear interpolations along a prediction direction by utilizing the neighboring reference pixels and the first half coded pixels. To alleviate the encoder computational load, [?] proposed an adaptive mode decision algorithm based on texture complexity and direction for HEVC intra prediction to filter out unnecessary coding block. The original redundant mode candidates for each PU are reduced according to its texture direction. In [?], to solve the two issues: the incoherence caused by the signal noise or the texture of other objects, and that the nearest reference line usually has worse reconstruction quality in block-based video coding, it proposed an multiple-line-based intra-prediction scheme to improve coding efficiency. Besides the nearest reference line, further reference lines are also utilized. The further reference lines with a relatively higher quality can provide potentially better prediction. At the same time, the residue compensation is introduced to calibrate the prediction of boundary regions in a block when we utilize further reference lines.

block illustration a.pdf block illustration a.bb

(a)

block illustration b.pdf block illustration b.bb

(b)

Fig. 3. Example of quadtree-based block partitioning with high textured content in a relatively flat block overall of RaceHorses (a) expected partitioning result in HEVC (b) expected partitioning result in proposed L-shaped block partitioning method

In these works, though SAP [?]-[?] can solve the problem of the long distance between reference samples and predicted samples, the RD cost is calculated based on the whole block because only one prediction mode is allowed in one block, which still hindering the block partitioning process from generating the most appropriate result for rich texture area. What's more, the SAP series of algorithms cannot be used in the lossy compression because the sample by sample method results in the diability in reconstruction, which is of great importance in lossy mode in HEVC.

B. QUADTREE-BASED BLOCK PARTITIONING IN HEVC

HEVC expands the 16 × 16 macroblock size used by the H.264/AVC standard to 64 × 64, which enables regions with simple textures or relatively static regions to be predicted and encoded in larger blocks, saving partitioning information and mode information overhead, so as to improve the coding efficiency . HEVC proposes the concepts of Coding Tree Unit (CTU) and CU to replace macroblocks, and adopts the quadtree-based block partitioning method to perform more accurate block partitioning result matching on different texture details. For area with rich textures, using smaller blocks for prediction can get smaller prediction residuals, but will result in more partitioning and mode information; regions with simple textures use larger blocks for prediction, which can save a certain amount of additional information, but at the same time the prediction distortion rises. This problem becomes more serious in regions farther away from the reference samples, like the block boundary. Therefore, we need to find a balance between more additional information and better prediction accuracy. The criterion used in HEVC is the rate- distortion optimization (RDO) between the prediction accuracy and the bit cost. The RD cost function (Jmode) used in HM is evaluated as follows [?]

$$J_{mode} = B_{mode} + \lambda_{mode} \cdot SSE \tag{2}$$

Where B_{mode} specifies bit cost to be considered for mode decision, which depends on each decision case. Sum of Squared Error (SSE) is the average difference between

reconstructed CU and the matching blocks, λ_{mode} is the Lagrange multiplier. In lossless case, the function will be

$$J_{mode} = B_{mode} \tag{3}$$

because no more distortion is produced when bypassing transform, quantization, and in-loop filters. In a block, RDO process is used to search for the best prediction mode (with the minimum RD cost) and the best block partitioning result.

In the process, a frame is first divided into several CTUs of the same size. Each CTU is the largest coding unit where RDO process is performed from the bottom to the top to determine the final block partitioning result. ??(a) shows an example of a CTU after block partitioning process. Each square represents an independent CU/PU (the two concept of CU and PU in this paper will not cause any obstacles to understanding, they all refer to a block to be predicted, coded and transmitted). HEVC uses a split flag to indicate whether a block is split to the next depth. After the block partitioning process, each CU performs prediction independently within itself.

It can be observed that under the quadtree-based block partitioning structure, since each block performs prediction independently, joint prediction is not allowed in the sub blocks under the same parent block even if they uses the same prediction mode. This causes unnecessary mode information. As shown in ??, due to the high textured content in a very small area in the lower right corner of the current block, the entire CTU is divided into 16 blocks The number may be, 12 or even smaller if we can merge these sub blocks together. This situation directly leads to a decline in compression performance.

Because of the high computational complexity in the implementation of quadtree-based block partitioning in HEVC, some works so far have tried to simplify the partitioning procedure. Cao et al. [?] proposed a short distance intra prediction (SDIP) based on the quadtree block partitioning structure, improving the intra prediction accuracy by splitting a coding unit into nonsquare units for coding and reconstruction. To decrease the coding complexity, memory access, and power consumption, which go against its widely applications, especially for ultra-high definition and/or mobile video applications, [?] proposed approaches consisting two stages of support vector machine-based fast INTRA CU size decision schemes at four CU decision layers to early terminate CU splitting or early skip checking the current CU depth. [?] allowed for significant reduction in computational complexity with small degradations in RD performance containing two complementary steps: 1) early CU split decision and 2) early CU pruning decision. The early CU splitting and pruning tests are performed at each depth level to avoid unnecessary calculation of RD cost. Choi et al. [?] proposed a new bottom-up-based block partitioning method called split-and-merge. This method splits an image into multiple square blocks and merges them into nonsquare blocks to exploit the dependence between split blocks. Moreover, a modification of the conventional intra prediction and transform is employed for nonsquare blocks. [?] proposed a coding framework supporting hierarchical splitting with binary and ternary trees and flexible coding order representations. The proposed compression scheme provides significantly higher compression capability than the state-of-the-art HEVC/H.265 standard for Standard Dynamic Range (SDR) category while maintaining complexity acceptable for emerging applications.

In other respects, [?] analyzed the problem with the current lossless coding scheme and propose a mode-dependent template (MD-template) based method for intra lossless coding.In order to achieve higher transform coding gains with relatively low-complexity implementations, [?] propose a joint separable and non-separable transform, which is named Enhanced Multiple Transform (EMT), applying multiple transform cores from a predefined subset of sinusoidal transforms, and the transform selection is signalled in a joint block level manner. [?] focused on CNN technology joining with image restoration to facilitate video coding performance, and propose the content-aware CNN based in-loop filtering for HEVC. [?] redesigned the intra mode coding method based on both short and long range correlations, as the existing approaches based on local content correlations cannot always effectively capture the most probable mode. And it achieved unified content adaptive coding and is applicable across different video content.

However, most works above are performed under the premise of the structure of binary, ternary or quad trees, and do not really solve the above problem described in Part B.

III. PROPOSED LOSSLESS INTRA-PREDICTION

As analyzed in Section II, two reasons motivate us to conduct this study: 1) The block-based prediction in HEVC results in inaccuracy of the prediction on the edge far away from the reference line of the block thus cannot accurately match the rich texture and 2) the generation of mode information redundancy because sub blocks under the same parent block cannot be predicted jointly.

In this section, we first introduce two proposed methods in details and then we present a merge method to maximize their gain.

A. L-SHAPE-BASED ITERATIVE PREDICTION (L-IP)

To overcome the first issue mentioned above, an L-shape-based Iterative Prediction (L-IP) method for HEVC lossless intra-prediction is provided. The core idea is to split a block into a series of close neighboring 1-D L-shapes and each L-shape prediction is performed by recursive method with previous neighboring L-shape as reference samples from top left to bottom right.

For a block of size N $\times$ N, we split it into N-1 L-shapes and one pixel (at the bottom right corner). As is shown in ??(a), the first L-shape, L1, is predicted in the same way as HEVC, using the decoded and reconstructed

Fig. 4. (a) Proposed L-shape-based iterative prediction process stopping at the last L-shape (b) Schematic diagram of different projection offset of neighboring prediction mode

TABLE I
residual reduction of L-IP

Class	Sequence	4 × 4		8 × 8		16 × 16	
		HEVC	L-IP	HEVC	L-IP	HEVC	L-IP
Class B	BasketballDrive	30.01	26.54	130.94	116.08	536.56	401.07
	Kimono	27.39	20.58	100.63	86.52	371.53	291.05
Class C	PartyScene	109.97	98.62	463.98	316.04	1067.07	860.59
Class E	Johnny	27.85	17.23	82.54	60.19	230.74	164.25
Class F	SlideEditing	163.6	135.32	337.67	248.71	1464.68	560.35
AVG reduction percentage		20.43%		22.14%		31.36%	

TABLE II
partitioning result comparison of L-shape-based iterative prediction (L-IP) and block-based prediction (BP)

Class	Sequence	4 × 4		8 × 8		16 × 16		32 × 32	
		L-IP	BP	L-IP	BP	L-IP	BP	L-IP	BP
Class A	PeopleOnStreet	19.27%	90.24%	5.58%	8.24%	28.21%	1.45%	46.94%	0.07%
	Traffic	14.93%	87.80%	4.22%	11.46%	29.50%	0.70%	51.35%	0.05%
Class B	BasketballDrive	48.44%	63.73%	22.78%	26.49%	16.30%	9.36%	12.48%	0.43%
	BQTerrace	42.09%	72.62%	11.98%	15.54%	19.79%	7.61%	26.14%	4.22%
AVG reduction/increment percentage		-47.42%		-4.29%		18.67%		33.04%	

samples of the nearest neighboring blocks (L0) as reference samples. Other L-shapes with m pixels use the decoded and reconstructed pixels of the closest neighboring L-shape which has m+2 pixels as reference line. Except for the reference samples, the generation of the predicted value is almost the same as in HEVC which calculates the linear interpolations of two reference samples neighbor to the intersection of the projection line and the reference line:

$$P^{Lj}_{(x,y)} = ((32-w)\cdot R^{Lj-1}_{(0,i)} + w \cdot R^{Lj-1}_{(0,i+1)} + 16) >> 5 \quad (4)$$

where the subscript, (x, y), represents the coordinates of the pixel in the current L-shape, which defines the top-left pixel as $(0, 0)$, and the superscript, Li, represents the number of the L-shape. Unlike in HEVC, we just need to pad (copy) one pixel at the two ends of each reference line like $R^{L1}_{(0,N)}, R^{L2}_{(0,N-1)}$ because as shown in ??(a). The maximum distance between the projection position and the edge of the reference line will be less than one pixel wide.

We can also see clearly from ?? and ?? that the distance between current position (yellow dots) and the reference sample (gray dots) in block-based prediction in HEVC is becoming longer while the position is closer to the right and the lower boundary. But in the proposed L-IP, the distance won't change with the position under a apecific prediction angle.

To demonstrate that the L-IP can reduce the residual energy, we provide some statistics of the average residual obtained by the two prediction methods in the case of the same block size. We study the 1^{st} frame of of Class B sequences (BasketballDrive and Kimono), Class C sequence (PartyScene), Class E sequence (Johnny) and Class F sequences. ?? shows that the average prediction residual is reduced by 20.43% in blocks of size 4 × 4, 22.14% in blocks of size 8 × 8 and 31.36% in blocks of size 16 × 16 (The number of blocks of size 32 × 32 is very small thus is not statistically significant).

After the process of L-IP, The pixels on the same L-shape will share the same prediction mode, so there are N prediction modes for a block of size N × N. Obviously, to encode and transmit these data directly will be an extra cost. Based on the strong correlation between adjacent pixels and blocks in the image picture, we can reasonably assume that in most cases, the prediction modes of adjacent L-shapes should be the same or only varies in a quite small range. "Mode Residual" is thus introduced to eliminate most redundancy in increased number of mode information due to iterative prediction method. It refers to encoding and transmitting only the variation of these N modes. In addition, it is easy to find that in HEVC, when the size of the block is small enough (for example, 4 × 4), the block-based prediction is accurate enough. If we still perform L-IP until the last L-shape, even if we can provide much more accurate prediction, the gain of that may be offset by the redundancy due to extra mode information, and it may even leads to worse overall performance. After statistical experiments on the HEVC test sequence, we found that keeping a base block in L-IP as 4 × 4 can save unnecessary expenses and provide the best performance. In the base block we still adopt L-IP except that there is only one prediction mode in the base block. In other words, every L-shape use the same prediction mode to generate the predicted value and the RDO process is also block-based. In this way, we can inhibit the generation of mode residuals as far as possible while ensuring the prediction accuracy.

Another improvement aim at L-IP is the number reduction of the prediction modes which is reduced from

process.pdf process.bb

Fig. 5. Block Partitioning result (a) Quadtree-based in traditional HEVC (b) Proposed l-shaped (c) Overall process of block partitioning from the block of size 8 × 8 to 32 ×32

35 modes to 8 modes. As is shown in ??(b), we can see that the offset of the projection between the neighboring angle mode is becoming smaller while the distance between the reference sample and the predicted sample. So using such high-density distributed angle modes is unnecessary for L-IP and will cause extra mode information. And the experimental data shows that the number reduction enabled in L-IP can even bring over 1% bit-rate saving.

Furthermore, in addition to the increase in prediction accuracy due to the shorter distance to the reference sample, there is another benefit that can be expected. For block-based prediction, a block can only have one certain prediction mode, this will results in an increase in the proportion of small blocks in the result of the block partitioning result for regions with rich textures. In L-IP, every L-shape can have its own separate prediction mode so that it can get more accurate prediction for rich textures within a larger block more accurately. This characteristic should be manifested as a large increase in the number of large blocks in the block partitioning result. It has also been verified in experiments shown in ?? (The partitioning results in row 3-6 denote the percantage of the pixel numbers of all the N × N blocks in pixel numbers of the whole frame). The proportion of blocks of size 16 × 16 and 32 × 32 is increased by 18.67% and 33.04% while that of size 4 × 4 and 8 × 8 is decreased by 47.42% and 4.29% respectively. Intuitively, with more larger blocks, more partitioning and mode information redundancy is eliminated, which can further help improve the performance of L-IP.

B. L-SHAPE-BASED BLOCK PARTITIONING (L-BP)

As mentioned in the previous section, sub blocks under the same parent block can't be predicted jointly in traditional HEVC algorithm. Once there is high textured content in the current block, there is a high probability that it will be split into 4 sub blocks. Extra mode information will be generated in those sub blocks that don't contain rich textures. ?? shows this situation. Because some high textured contents exist at the lower-right corner, the whole block of size 64 × 64 is split into 16 blocks. This can be worse especially when using L-IP method because one more block of size N × N means N-2 more mode residuals. Considering this situation and the shape characteristics of L-IP, we propose an L-shape-based Block Partitioning (L-BP) method. In this way, we can ensure the retention of the larger blocks while there has a high textured content, so as to fully remove the redundancy of the mode information and further maximize the gain of L-IP. The so-called L-shaped block partitioning is to treat a block as two parts: the L-shaped part and the reserved block part. This process is equivalent to merge the three of the sub blocks together as the L-shaped block for joint prediction after the block partitioning process using traditional HEVC method. Since the high textured content mentioned earlier may occur anywhere in the current block, the reserved block is allowed to be located on 4 corners in this algorithm, namely upper left, upper right, lower left, and lower right.

For the reserved block, we expect to retain the block partitioning results of previous depth level, otherwise repartitioning will cause huge amount of unnecessary overhead. A contradiction occurs here that during the basis for judging whether a block should be split in HEVC: RDO process. B_{mode} in (??) contains the bit cost of the mode information, the partitioning information and the bit cost of transmitting the encoded residual data, so the mode and partitioning information generated during block partitioning will affect the bit cost after encoding, thereby changing the final RD cost. Once the RD cost changed, the final partitioning result will also change.

In HEVC, mode information will not be transmitted directly. For pictures and videos, the high correlation between neighboring blocks results in a high probability that the intra prediction modes of adjacent blocks will be the same or similar. Therefore, the HEVC standard construct a modes list including two most probable modes (MPMs) and additional probable intra modes when intra mode coding [?]. There are 3 candidate modes which are used to store the prediction modes of spatially nearby blocks (the above block and the left block).

Based on the analysis above, if we want to retain the results of the previous depth level, once the prediction modes of the blocks nearby changes, the MPM will also change, resulting in the different value of the RD cost and finally leads to different partitioning results of the reserved block. This may even lead to wrong mode information decoded by the decoder and cause serious distortion. This is equivalent to overturning the original partitioning result which we want to retain.

For these reasons, we are to ignore the inaccuracy in RD cost, re-construct MPMs and recode the mode information after the block partitioning result is completely determined. Because the main factor that determines the result of the block partitioning is the texture characteristics of the image in the current block, it is reasonable to speculate that the impact to the block partitioning result of directly ignoring the inaccurate RD cost caused by the mode information can be ignored.

In order to verify the feasibility of this method, we compare the RD cost of the blocks of size 4 × 4 and 8 × 8 with the recalculated RD cost after the reserving process is completed. Experimental result shows that for the reserved blocks, the difference between the two RD cost is ±0.27% for 4 × 4 blocks and ±0.29% for 8 × 8 blocks in average, which can almost be ignored.

Fig. 6. Flowchart of the proposed L-BPIP in HEVC lossless intra-prediction

cases.pdf cases.bb

Fig. 7. Cases of a block to be compared in the RDO process

Fig. 8. Two special cases of L-IP in L-BPIP

The overall process of block partitioning uses the same bottom-up approach as in HEVC. The method we proposed is equivalent to adding a splicing step on the basis of the traditional block partitioning result. ?? shows the splicing process. ??(a) shows the partitioning result in traditional HEVC; ??(b) shows the L-shaped block partitioning result after the splicing process; ??(c) shows the bottom-up comparison process. Starting from the smallest block of size 4×4 (case 4×4 is omitted in the figure owing to space constraints), the comparing process is performed in each level of depth, and finally determine the block partitioning result of the entire CTU.

C. L-SHAPE-BASED BLOCK PARTITIONING AND ITERATIVE PREDICTION (L-BPIP)

According to the aforementioned analysis, combining the two methods together can maximize their advantages. There- fore, we propose a merge method that combines L-BP and L-IP called L-shape-based Block Partitioning and Iterative Prediction (L-BPIP) algorithm to replace the one in traditional HEVC intra-frame lossless compression. We use the same criterion of block partitioning: RDO process. For any input CU, the RDO process will be performed under 11 cases (shown in ??) and the best case (with minimum RD cost) is chosen to construct the final partitioning result.

Since there are 11 situations to be compared, directly transmitting will require 4bit-wide data. We have found that in these cases, three account for most of the proportions ("split to next depth" accounts for 61%, "lower-right reserved BP" accounts for 9% and "lower-right reserved L-IP" accounts for 9%). So we decide to use variable length coding (VLC) to save the redundancy in partitioning information. ?? shows a flowchart of the proposed overall algorithm, where Rd_* means different RD cost to be compared during the whole process, Rd_nons means the RD cost of non-split blocks, Rd_split means the one of four sub blocks under the current block. Rd_res means the RD cost of blocks have reserved blocks in it, Rd_blk means the RD cost of blocks using block-based prediction while Rd_L_sum means the sum of the RD cost of L-shapes and base blocks using L-IP, Rd_res_i means the RD cost of four cases of reserving-based intra prediction.

During the L-IP performed to the L-shaped block in this merge method, in order to uniform the overall process, we adopt the same prediction order from the upper-left L-shape to the lower-right L-shape for all four reserved cases. Except that the case in the upper-left reserved which predict the reserved block first and the L-shapes later, and the case in the lower-right reserved which predict the L-shapes first and the reserved block later, other three cases may have special prediction order in L-IP to solve the problem of missing reconstructed reference samples, which are discussed here:

1) Upper-left reserved: first predict and reconstruct the re- served block, and then perform L-shape-based iterative prediction to the remaining pixels.
2) Upper-right or lower-left reserved: the prediction order is L1 $\rightarrow$ L2 $\rightarrow$ L3 $\rightarrow$ L4 $\rightarrow$ B1 $\rightarrow$ B2.
3) Lower-right reserved: the prediction order is L1 $\rightarrow$ L2 $\rightarrow$ L3 $\rightarrow$ L4 $\rightarrow$ reserved block.

The schematic diagram of the prediction process is shown in ??.

At the decoding side, we have the same order as the encoding side, we use the decoded and reconstructed result of it as reference samples and do iterative prediction to the next neighboring L-shape. For the special cases above, the algorithm iterates in the order: L1 $\rightarrow$ L2 $\rightarrow$ L3 $\rightarrow$ L4 $\rightarrow$ B1 $\rightarrow$ B2.

In this prediction and decoding order, we can make sure that before every block or L-shape is predicted, the reference samples it needs are already reconstructed.

IV. IMPLEMENTATION AND OPTIMIZATION

In the implementation of RDO, HEVC always encodes the residual coefficient in the size of 4×4 for all block size when transmitting, so the RD cost can be considered as the sum of RD costs of 4×4 units. For example, the RD cost of the 8×8 block is the sum of B1, B2, B3 and B4 though predicted in a whole block in ??. For HEVC uses context-based adaptive binary arithmetic coding (CABAC) to encode the data need to be transmitted, the context-based characteristic of it leads to different results of RD cost if the coding order is changed.

As ?? shows, in the case of upper-right reserved-BP, the coding order for the L-shape block is B1 $\rightarrow$ B3 $\rightarrow$ B4. B2 is missing because the reserved block needs to be processed separately. This results in the changing context of B3, and different encoding result and RD cost at the meantime. So in former L-BPIP, the L-shape block of four reserved cases need to be recoded to ensure the correct RD cost. This process is actually ineffective and it increases the encoding time. The reason is that the L-shape block and the square block use the same reference sample during block-based prediction, which leads to the same predicted value for the pixels in the same position and it can be reasonably assumed that the difference of the RD cost due to the changing context can be ignored.

So we proposed the scheme to reduce the encoding time. During the RDO process of the case that having L-shape

$$P_{(x,y)} = \begin{cases} min(R_{(x,y-1)}, R_{(x-1,y)}) & if R_{(x-1,y-1)} \geq max(R_{(x,y-1)}, R_{(x-1,y)}) \\ max(R_{(x,y-1)}, R_{(x-1,y)}) & if R_{(x-1,y-1)} \leq min(R_{(x,y-1)}, R_{(x-1,y)}) \\ R_{(x,y-1)} + R_{(x-1,y)} - R_{(x-1,y-1)} & otherwise \end{cases} \quad (5)$$

TABLE III
LOSSLESS CODING PERFORMANCE COMPARISON OF THE PROPOSED METHODS IN TERMS OF BIT-RATE DIFFERENCES WITH RESPECT TO HEVC ALL-INTRA UNDER LOSSLESS CONFIGURATION

Class	Sequence	L-IP	L-BP	L-BPIP			
		All	All	Y	Cb	Cr	All
Class A (2560 × 1600)	PeopleOnStreet	-11.29%	-1.78%	-14.03%	-13.86%	-11.43%	-13.42%
	Traffic	-11.46%	-1.89%	-14.30%	-9.60%	-13.98%	-13.26%
Class B (1080p)	BasketballDrive	-4.20%	-2.28%	-2.74%	-3.72%	-9.90%	-4.30%
	BQTerrace	-8.57%	-1.80%	-9.32%	-7.40%	-7.84%	-9.12%
	Cactus	-3.74%	-1.15%	-3.76%	-2.84%	-5.34%	-3.92%
	Kimono	-7.71%	-1.57%	-6.74%	-13.72%	-21.14%	-9.07% (-7.57%)
	ParkScene	-6.58%	-1.48%	-7.50%	-9.36%	-10.43%	-7.95% (-8.27%)
Class C (WVGA)	BasketballDrive	-5.51%	-1.96%	-2.81%	-2.06%	-4.63%	-3.76%
	BQMall	-6.17%	-1.46%	-5.80%	-7.06%	-10.91%	-6.55%
	PartyScene	-5.58%	-1.42%	-5.27%	-6.74%	-8.30%	-5.71%
	RaceHorses	-8.10%	-1.46%	-8.14%	-14.89%	-15.70%	-9.21%
Class D (WQVGA)	BasketballPass	-12.36%	-2.62%	-13.02%	-16.15%	-17.76%	-13.79%
	BlowingBubbles	-5.47%	-1.05%	-4.89%	-8.35%	-11.11%	-5.42%
	BQSquare	-5.27%	-1.54%	-4.37%	-6.20%	-8.33%	-5.05%
	RaceHorses	-8.51%	-1.17%	-8.58%	-13.80%	-14.54%	-9.38%
Class E (720p)	FourPeople	-12.51%	-3.52%	-11.43%	-23.62%	-22.93%	-14.14%
	Johnny	-10.97%	-4.11%	-9.15%	-20.92%	-20.63%	-12.29%
	KristenAndSara	-11.86%	-4.40%	-10.44%	-21.16%	-20.91%	-13.24%
Average of Class A - Class E		-8.10%	-2.04%	-7.90%	-11.19%	-13.10%	-8.87%
Class F (screen content)	BasketballDrillText	-6.07%	-1.96%	-3.39%	-3.30%	-5.83%	-4.50% (-6.38%)
	ChinaSpeed	-17.15%	-4.75%	-19.37%	-15.40%	-16.22%	-18.56% (-14.42%)
	SlideEditing	-13.57%	-2.64%	-16.86%	-8.24%	-9.54%	-15.49% (-11.80%)
	SlideShow	-22.26%	-5.43%	-25.25%	-26.96%	-26.22%	-26.42% (-18.92%)
Average of Class F		-14.76%	-3.70%	-16.22%	-13.48%	-14.45%	-16.24% (-12.88%)
Average of All		-9.31%	-2.34%	-9.42%	-11.61%	-13.35%	-10.21%
Encoding time		125%	101%	129%			
Decoding time		103%	100%	103%			

Results in parenthesis [?] indicate the benefits of SAP-E [?] with respect to HEVC all-intra under lossless configuration.

Fig. 9. The RD cost calculation order in the case of upper-right reserved-BP

blocks and choose block-based prediction, we directly use the four RD costs generated in the block-based predicted CU where the L-shape block is located and calculate the sum of three of them (in ??, the three is the RD cost of B1, B3 and B4, they are generated during the 8 × 8 block is predicted as a whole using block-based prediction) according to which position the reserved block is as the RD cost of the L-shape block. This optimization can save the block-based prediction and encoding process for L-shape blocks.

Furthermore, to maximize the benefits of reducing the distance between the reference samples and predicted sample, we apply SAP-E [?] in planar prediction mode, in which the predicted value is calculated as (??) [?].

And sample by sample method is performed to horizon (mode 10) and vertical (mode 26) prediction mode.

V. EXPERIMENTAL RESULTS

To verify the effectiveness of the proposed method, we perform the experiment in the HEVC reference software HM-16. We take the whole range of HEVC test sequences as our test sequence. There are six classes containing five natural video classes and one screen content class: Class A (2560 × 1600); Class B (1080p); Class C (WVGA); Class D (WQVGA); Class E (720p); Class F (screen content). The experiment strictly follows the HEVC common test condition (CTC) [?]. The bit-rate gain is used to assess the coding performance of the proposed methods, where a negative number indicates bit-rate saving. Since our methods are designed for lossless intra-coding, we only test the lossless All Intra (AI) configuration in HM. ?? shows the coding performance of the proposed three intra-frame lossless compression for the three luma and chroma components and the whole gain compared with the original coding method in HEVC for each sequence under lossless AI. The encoding and decoding time are tabulated in the last two lines. We use the following ΔT to calculate the

Fig. 10. Block partitioning result comparison of the proposed L-BPIP with respect to HEVC all-intra under lossless configuration

encoding or decoding time difference,

$$\Delta T = \frac{T_{proposed}}{T_{anchor}} \times 100\% \tag{6}$$

where $T_{proposed}$ means the encoding time or decoding time of the proposed method, T_{anchor} means the encoding or decoding time of the original HEVC method in HM-16.

A. L-shape-based Iterative Prediction (L-IP)

The experiment result for L-IP is shown in the third column of ??. For L-IP, the maximum bit-rate saving is 12.51% for FourPeople in Class A - Class E, and 22.26% for SlideShow in Class F, with an average of 8.10% for Class A - Class E and 14.76% for Class F. For computational complexity, L-IP is observed to increase the encoding time by 25%, with almost the same decoding time.

B. L-shape-based Block Partitioning (L-BP)

As shown in the fourth column of ??, the maximum bit-rate saving of L-BP is 4.40% for KristenAndSara in Class A - Class E, and 5.43% for SlideShow in Class F, with an average of 2.04% for Class A - Class E and 3.70% for Class F. For this method, both the encoding and decoding time have barely changed.

C. L-shape-based Block Partitioning and Iterative Prediction (L-BPIP)

Column 5-8 of ?? shows the gain of L-BPIP which has the best performance of the three proposed methods. The maximum bit-rate saving is 14.14% for FourPeople in Class A - Class E, and 26.42% for SlideShow in Class F, with an average of 8.87% for Class A - Class E and 16.24% for Class F. L-BPIP is observed to increase the encoding time by 29%, with almost the same decoding time. This level of increase in encoding time is objectively acceptable for this degree of computational complexity.

D. Comparison to SAP-E

In HEVC intra-frame lossless compression, The SAP series of algorithms have the highest benefits in the published literature in our knowledge. Among them, SAP-E [?] perform the best efficiency. Let us recall that though SAP series of algorithms use sample by sample method in prediction, in RDO process, they still take blocks as units, which affect the accuracy of the block partitioning. L-BPIP use several L-shape and allow them to have different prediction mode to avoid the problem. In [?], it test only part of HEVC test sequences: Class B-ParkScene,Class B-Kimono, and Class F. It can be seen in ?? that the bit-rate saving of L-BPIP is increased by 0.59% and 3.36% with comparison to SAP-E in the average benefit of Class A - Class E and Class F.

E. Further analysis

From the statistics above, we can observe that compared with L-IP and L-BP, the bit-rate saving of L-BPIP is increase by 0.89% and 7.87%, which proves the significance and necessity of combining L-IP and L-BP. In the meantime, compared with the block partitioning result in traditional HEVC method, the proportion of 16×16 and 32×32 blocks of L-BPIP is increased by 25.82% (shown in ??), which confirms our conjecture that the intra-frame lossless compression coding using the combination of L-IP and L-BP can increase the number of larger blocks, which provides L-IP with favourable conditions. It is easy to imagine that the larger the current block is, the greater the gain brought by the use of L-IP will be, because it can save a considerable part of mode information generated in the traditional method, while ensuring the same or even better accuracy of the prediction.

Further more, it is worth noting that, although L-BPIP needs to compare 11 cases during RDO process,

the encoding time only increases by less than 1 time. As is analyzed in section III, part B, the RD cost of 5 cases concluding Block-based prediction (BP) among all the 11 cases can be directly obtained from the result of the previous depth level because we retain the partitioning results instead of repartitioning. That's the main reason of the reducing encoding time. Another reason is that in RDO process,we use the bit cost of the original prediction residual instead of the one of the encoded prediction residual.

VI. conclusion

In this paper, we proposed a new prediction method and a new block partitioning method to increase the coding efficiency for HEVC intra-frame lossless compression. First, we design an L-shape-based iterative prediction to decrease residual instead of traditional block-based prediction. Secondly, an L-shape-based block partitioning method is presented to take full advantage of the new prediction method and reduce mode information redundancy. The experiment results show that the proposed new methods can provide up to 14.14% bit rate reduction with an average of 8.87% and we adopt some implementation optimization to speed up the whole prediction and RDO process,increasing the encoding time by only 29% when it is enabled, which is acceptable for process of this complexity.