Multiple Line-based Intra Prediction for High Efficiency Video Coding

Jiahao Li, Bin Li, Member, IEEE, Jizheng Xu, Senior Member, IEEE, and Ruiqin Xiong, Member, IEEE

Abstract—Traditional intra prediction usually utilizes the nearest reference line to generate the prediction block when only considering the strong spatial correlation. However, this kind of single line-based method is not always a good solution because of two reasons. One is that the reconstruction quality of pixels in different positions of a block vary from each other if a block has been quantized in the transform domain. In general, the boundaries (especially block corner) have a worse quality than inner regions for smooth regions. The other is the existence of signal noise and object occlusion. Few of noise or occlusion on the nearest reference line may lead to a large prediction deviation. Because of the two reasons, this paper proposes to utilize multiple reference lines to improve the intra prediction. Besides the nearest reference line, further reference lines are also used to generate the prediction, in where the residue compensation and weighted prediction are introduced. The advantages of the proposed method include that it probably provide more accurate prediction from further reference lines with a higher quality. The other advantage is that the proposed method can reduce the influence of noise or occlusion on the reference line to improve the coding efficiency. In addition, this paper also designs several optional acceleration algorithms when considering the encoding complexity. Experimental results show that for all intra configuration, the bit saving of the proposed fast scheme is $1.9\%$ on average, and up to $3.2\%$, with increasing the encoding time $115\%$.

Index Terms—High Efficiency Video Coding, quality analysis, noise and occlusion analysis, multiple reference line, residue compensation.

I. INTRODUCTION

The newly published High Efficiency Video Coding (HEVC) can provide a similar perceptual quality with about $50\%$ bitrate saving compared with its predecessor H.264/AVC [2]. The first edition has been finalized in January 2013, also known as HEVC version 1.

HEVC can be the state-of-the-art video compression scheme owing to the introduction of many new technologies and improvements based on hybrid coding framework, including several enhancements for the intra prediction at the same time. For example, the number of angular direction has been extended to 33 in HEVC, and this kind of fine-grained direction can provide more accurate prediction than the previous H.264/AVC standard, in where the number is only 8. To solve the discontinuities caused by the plane prediction mode in H.264/AVC, the improvement is also made, and the plane mode is promoted in the form of planar mode in HEVC [3]. In H.264/AVC, the coding unit size is fixed on 16x16 macro block. By contrast, HEVC introduces the concepts of coding unit (CU), prediction unit (PU), and transform unit (TU). For intra coding, PU carries the prediction information such as intra direction, and TU is the basic intra prediction unit. All TUs in a PU will share the same prediction information, in where the size of TUs varies from 32x32 to 4x4. Small TU will shorten the spatial distance between prediction pixel and reference pixel, then reduce the prediction error under the basic assumption that the spatial correlation will be stronger with the distance shorter. As the partition of PU and TU is also decided by the rate distortion optimization like CU, this kind of flexibility will improve the coding efficiency. In addition, HEVC intra coding enables discrete sine transform (DST) for 4x4 luma TU to further reduce the residue energy.

In HEVC intra coding, the generation of angular prediction still follows a copying-based method like H.264/AVC. A two-tap linear interpolation by the reference pixels is as the predictor, in where the linear weigh is the inverse proportion of distance between the two adjacent reference pixels. However, this kind of two-tap linear weight may not be the optimum as it only assumes a simple model that the visual content follow a pure direction of propagation. Actually the optimum weight which leads to the minimum mean square error (MMSE) can be obtained by the statistic of current block and reference pixels. In [4], a position-dependent filtering method based on MMSE is proposed, in where each position has its own weights under different block sizes and directions. Nevertheless, a huge weight table need be stored in the decoder, and it is not welcome. To avoid the using of large weight table, many works [5,8] refine the intra prediction by modeling that the correlation between adjacent pixels follow a first order 2-D markov process, in where each pixel is predicted by linearly weighing several adjacent pixels. The weights in markov model still need to be off-line trained just like the method [4], and has not a good adaptivity when encoding new sequences. Another typical technology used for improving intra prediction is the image inpainting. In [9,10], an edge-based inpainting method is proposed. The methods [11,12] make use of the partial differential equations based inpainting technology. The total variation based inpainting is also utilized in [13]. Although the intra prediction with inpaiting technology can generate a more accurate prediction than directional copy-based method, it is often with a high computational complexity so that it is not popular in practical codec. For this purpose, Lai et al. propose an error diffused intra prediction algorithm for HEVC, and show it outperforms the inpainting technique, especially.
with much lower computational complexity [14]. In addition, a novel method proposes to encode half pixels of a block firstly, and the remains are reconstructed by interpolations utilizing the neighboring reference pixels and the first half encoded pixels [15]. The coding gain mainly comes from that the prediction distance is shorten.

However, the solutions [4]-[15] including the intra prediction in HEVC standard only utilize the nearest reference line to generate the prediction when only considering the strong correlation. Actually some further regions are also can be utilized as there may exists similar content as the predictor. One famous work is the template matching [16] which search the non-local similar content by using the neighbor reference pixels as indicator. It performs well on the situation there are many repeated patterns. As there is a matching procedure, it is with high complexity, especially for the decoder.

In this paper, we propose to take advantage of the local further regions, namely the further reference lines. The motivation of this paper includes two parts. One is that the pixels in different positions of a block have the different reconstruction quality under lossy coding condition. This is because the residues are quantized in the transform domain. The unequal frequency quantization error will lead to different spatial quantization error [17]. In general, the boundary of a block has a larger quantization error (i.e. worse quality), especially on the corners. So the nearest reference line utilized by traditional intra prediction solutions are often with a worse quality, which is not we expected. Another motivation is the original signal noise and object occlusion. If only one single reference line is used, the encoder will not have a good robustness on noise/occlusion tolerance because the actual prediction will have a obvious deviation when few reference pixels are noised or there exists object occlusion. According to the multiple hypotheses of signals, the further reference lines probably provide a better prediction. For the two motivations, this paper proposes a multiple line-based intra prediction for HEVC. The prediction generated from further reference line will compete with the prediction generated from the nearest reference line in order to choose the best prediction for each block. When further reference line is used, a residue compensation procedure is introduced to further refine the prediction, followed by weighted prediction. In addition, this paper also designs several optional acceleration algorithms when considering the computational complexity. Experimental results show that for all intra coding, the bit saving of the proposed fast scheme is 1.9% on average, and up to 3.2%, with increasing the encoding time 115%. If ignoring the encoding complexity, the results also show that the benefit will continue to increase with the number of reference lines larger, even when the number is up to twelve. It should be noted that the proposed multiple line-based intra prediction can cooperate with other solutions which only use single reference line such as the works [4]-[15], then jointly optimize the intra prediction.

The rest of this paper is organized as follows. Section II reveal the motivation of the proposed method with detailed analysis. Section III introduces the proposed method including the basic framework, the residue compensation, weighted prediction and corresponding fast algorithms. Experimental results are presented in Section IV. Section V concludes this paper.

II. Analysis

This section is divided into two parts. We firstly analyze the reconstruction quality of different regions in a block. Then the signal noise and object occlusion influence for the line-based intra prediction is analyzed under lossless coding condition. The two parts are exactly the motivation of this paper.

A. Analysis of Reconstruction Quality

In most video/image lossy coding frameworks, the residue is quantized in the transform domain, and it introduces the frequency quantization error. As the transform and inverse transform are the linear operation, the spatial quantization error is the inverse transform of the frequency quantization error. Furthermore, from a statistical perspective, the relation between the variance of spatial quantization error and that of frequency quantization error also can be derived theoretically, and Robertson et al. [17] have present the relation for the discreet cosine transform (DCT). They conclude that the locations near the block boundary have a relatively higher spatial quantization error variance for smooth signals whose signal energy is mainly contained in the low frequency coefficients. On the contrary, for signals that contain significant high-frequency content, such as textured regions, the inner pixels of a block have a higher spatial quantization error variance.

In most cases, the smooth block take a larger proportion in a sequence when compared with the complex texture block, and it leads to that the error variance of boundary region is larger than that of the inner region according to the aforementioned conclusion. To verify this assumption, we calculate the variance of spatial quantization error by doing statistic. Fig. 1 shows the variance of spatial quantization error for different block sizes of the $1920 \times 1080$ Traffic sequence. The sequence is compressed by the HEVC reference software.
with all intra configuration and quantization parameter (QP) 37. From the figure, we can see that the most marginal region has the worst quality in a block, especially on the right-bottom corner. However, the most right column and the most below row exactly are the reference pixels used for traditional intra prediction. For this reason, it motivates us to utilize the relatively inner regions of a block to improve the intra prediction accuracy as they have an obviously better quality. It is noted that the 4x4 luma block is with DST in HEVC intra coding, and blocks with other sizes use DCT. Nevertheless, these blocks follow the similar characteristic from the quality distributions shown in Fig. 1

B. Analysis of Signal Noise and Object Occlusion

The signal noise is introduced during signal acquisitions. It not only harm the video quality, but also reduce the coding efficiency as it will decrease the prediction accuracy. In essence, the intra prediction in HEVC is a single line-based copying process with the assumption that the visual content follows a pure direction of propagation. If the nearest reference line is noised, the noise also will be propagated into the whole prediction block, and it will form an obvious noise stripe. Similarly, if the textures of the prediction block and further reference line are with continuous propagation, the occlusion on the nearest reference line will lead to that a large region of block has a deviation. The noise stripe and occlusion have a bad influence on the prediction accuracy, which is not we expected. However, according to the multiple hypotheses of signals, the further reference lines probably provide a better prediction with less noise, or they can break the constraint of occlusion. So the utilization of multiple reference lines will improve the robustness on noise/occlusion tolerance.

To verify the bad influence of signal noise and occlusion, we illustrate two examples in the Fig. 2. In the figure, the left block is the original block. The middle block is the prediction block generated by the nearest reference line. The right block is the prediction block generated by the further reference line. They are with lossless coding condition to exclude the influence of the quantization error in the reference line. Fig. 2 (a) shows the example of signal noise, in where the directions of the nearest reference line and further reference line are both 5. From the middle block, we can see the obvious noise stripe, and the noise source is indicated by the ellipse. Fig. 2 (b) shows the example of occlusion, in where the directions of the nearest reference line and further reference line are 7 and 6 respectively. From the middle block, we can see that a large region of the prediction block has a deviation with the left original block because of the existence of the occlusion on the nearest reference line. However, the further reference line can provide more better prediction according to the according to the multiple hypotheses of signals, as shown in the right block in the Fig. 2. Actually, in most cases, the signal noise and occlusion are hard to distinguish in a two or three pixel offset. Nevertheless, the multiple line-based scheme can handle with the two problems well, and we do not care the their existential form.

III. PROPOSED MULTIPLE REFERENCE LINE SCHEME FOR INTRA PREDICTION

Based on the analyses conducted on the previous section, this paper proposes a multiple reference line-based scheme to improve the coding efficiency. In the proposed method, not only the nearest reference line is used in intra prediction, but also the further reference lines are also made use of. By utilizing further reference lines with higher quality, the proposed method probably provide better prediction. According to the multiple hypothesis of signals, the multiple reference line-based intra prediction can also suppress the influence of noise and occlusion when compared single line-based method.

In this section, we firstly introduce how to generate the prediction block with the further reference line, as it is the basis in the proposed method. In the procedure of predicting the block with further line, some affiliated informations also can be obtained. To take the full advantage of these informations, we propose a residue compensation as an post processing for the prediction block. After the residue compensation, we will weight the prediction generated by the nearest reference line to further refine the prediction. Beside the coding efficiency, we also care about the computational complexity. For this reason, an alternative solution is designed, in where several acceleration algorithms are incorporated.

A. Prediction Generation

In the proposed scheme, the structure of the multiple reference lines is organized as shown in Fig. 3. Each reference line is indicated by an index $L_X$ (the nearest reference line is corresponding to index $L_0$). From Fig. 3 we can see that there exists an interval between the reference line $L_X$ and the prediction block, whose offset is $X$.

In the generation of prediction block, the further reference line takes the same rule with the nearest reference line. For the 33 angular directions, each pixel of prediction block is projected on the reference line $L_X$ along the direction, then using the interpolated value (at 1/32 pixel accuracy) as the
predictor. The average of the further reference line is as the predictor for the DC mode. In planar mode, each pixel is an average of two linear predictions (see [12] for details, and we only replace the nearest reference line with the further reference line). However, beside the origin prediction block, the pixels in the interval are also able to be predicted by the further reference line. This characteristic will help us design the post-processing, and the details will be introduced in the next subsection. So we will predict a larger block to support the post-processing, in where the size of the larger block equals the original prediction block size plus one. It is illustrated in Fig. 4. For example, we will generate a 5x5 block if current block to be predicted is 4x4.

It is noted 4N + 1 pixels will be utilized in generating the prediction block with the nearest reference line $L_0$ [3]. More generally, $4(N + x) + 1$ pixels will be utilized for $L_X$ as the existence of the interval. The extended pixels are illustrated in Fig. 5. For each reference line, the unavailable pixels padding and the reference pixel smoothing follow the same manner with the nearest reference line in the HEVC standard [18].

The reference line index $L_X$ will be transmitted into the bit stream as the decoder needs to know which reference line to predict the block. It is coded using a similar way to code reference picture index. For example, if there are four reference lines, we will use 0, 10, 110, and 111 to indicate $L_0$ to $R_3$ respectively. In the proposed method, the reference line index is implemented on CU level when considering that more overheads used for transmitting the index will be introduced to harm the coding efficiency if it is conducted on PU or TU level. So all PUs and TUs in a CU will share the same reference line index. For the same reason, the chroma components will reuse the downsampled reference line index of its corresponding luma component to improve the coding efficiency.

### B. Residue Compensation

As aforementioned in previous section, the interval region are also able to be predicted, as affiliated informations in utilizing the further reference line. In addition, the reconstructed pixels of the interval are also available if current block does not locate on the boundary of a picture. So the residue can be obtained by subtracting its “new” prediction from its reconstruction, where the new prediction means the prediction is generated by the further reference line. The residue of the interval can be utilized to improve the prediction of current block because we assume that the residue also have a spatial correlation in some degree.

This paper proposes to compensate the neighbor residue on the prediction block to suppress the discontinuities along block boundaries, which is introduced in the intra prediction. Considering the assumption that spatial correlation is stronger with the distance shorter, we only compensate the residue of the nearest reference line to the prediction block. This is the reason why we predict a larger block whose size equals the original blocks plus one in the previous subsection. To improve the efficiency, this paper designs several residue compensation strategies to adapt different intra modes, and they are described subsequently.

The perpendicular residue compensation is applied on the both of the above and left boundaries, in where the above rows of a prediction block are with vertical residue compensation and the left columns are with horizontal residue compensation. It can be formulated as:

$$
p_{j,i} = \begin{cases} 
p_j + w_j \times (r_{j-1,i} - p_{j-1,i}) & j < K \\
p_j + w_i \times (r_{j-1,i} - p_{j-1,i}) & i < K 
\end{cases}$$

where $p_{j,i}$ is the prediction value of each pixel, in where $j$ and $i$ are the row and column location index respectively. $r$ is the corresponding reconstruction value. It is noted that (0,0) represents the first pixel of original prediction block. $K$ is the parameter representing the number of compensated row/column. $w$ indicates the weight of the compensation. It is calculated as:

$$w_k = (A - k) / B \quad K \leq A, 0 \leq k < K$$

where $k$ is the index of row/column. It means that the pixel with a shorter distance to the boundary will have a larger compensation weight according to its stronger correlation with the residue. In the implementation, the $K$ is set as 3 for perpendicular compensation. $A$ and $B$ are the parameters, which are set as 3 and 4 respectively. Fig. 5 (a) illustrates the
compensation, in which the gradually changed color indicates the intensity of compensation weight. In (1), it is noted that the above-left part \((0 \leq i, j < K)\), denoted by the region within dotted lines in Fig. 5 (a)) will both perform the horizontal and vertical compensation. In the proposed method, the perpendicular compensation is used for the DC and planar compensation.

The vertical residue compensation is performed on the above rows of a prediction block. It is as:

\[
p_{j,i} = p_{j,i} + w_j \times (r_{-1,i} - p_{-1,i}) \quad j < K
\]

(3)

It deserves to be mentioned that the \(w_j\) will be adjusted according to the direction mode, and it is formulated as:

\[
w_k = (14 - \text{abs}(\text{dir} - \text{HOR}))(K - k)/64 \quad 0 \leq k < K
\]

(4)

where the \(\text{dir}\) is the direction mode index, and \(\text{HOR}\) is the horizontal mode index. For vertical compensation, \(K\) is set as 1 in the implementation. The vertical compensation is shown in Fig. 5 (b). In a similar way, the horizontal residue compensation acts on the left columns of a prediction block, which can be regarded as a transpose of vertical residue compensation. It is shown in Fig. 5 (c). In the proposed method, vertical residue compensation will be used for angular directions around horizontal direction (i.e. mode 7 \(\sim\) 13), and the horizontal residue compensation is with directions around vertical direction (i.e. mode 23 \(\sim\) 30).

In the proposed method, the parallel residue compensation will follow the same direction with the intra angular direction, it is as:

\[
p_{j,i} = p_{j,i} + w_{\min(i,j)} \times (r_\ast - p_\ast) \quad j > K \| i < K
\]

(5)

where the * is the operation of projecting the \((j, i)\) pixel on the residue of the nearest reference line along the angular direction. If the projected locates on a fractional position, the \(p_\ast\) and \(r_\ast\) is interpolated by the two adjacent pixels, same with the normal intra angular prediction. We take \(K\) as 3 for parallel compensation. The weight is same with that in (2). The parallel compensation is illustrated in Fig. 5 (d), and performed on angular directions around diagonal direction (i.e. mode 14 \(\sim\) 22).

In addition, we propose inversely parallel compensation for mode 2 and mode 34, and it is shown in Fig. 5 (e) and Fig. 5 (f) respectively. The formulation is as:

\[
p_{j,i} = \begin{cases} 
p_{j,i} + w_j \times (r_\ast - p_\ast) & 0 \leq j < K \& \text{dir} = 2 \\
p_{j,i} + w_i \times (r_\ast - p_\ast) & 0 \leq i < K \& \text{dir} = 34 
\end{cases}
\]

(6)

For mode 2, the above rows are compensated. It is noted that the * will project the pixel \((i, j)\) on the above reference row rather than the left reference column as there are two intersections with the nearest reference line along the intra direction. In a similar way, the left columns are compensated for mode 34, and the pixel \((i, j)\) is projected on the left reference column. In the implementation, \(K\) is set as 3 for inversely parallel compensation. The weight is same with that in (2).

To verify the benefit of residue compensation, we calculate the mean square error (MSE) between the prediction block and original block. Then we illustrate the MSE difference which is the MSE of the proposed multiple line-intra prediction minus that of the traditional single line-intra prediction, shown in Fig. 6. In the figure, the statistic is with all 16 \(\times\) 16 blocks in a picture of 3840 \(\times\) 2160 TrafficFlow sequence. The upper layer data is the proposed method without the residue compensation, and the lower layer data is with the residue compensation. From the upper layer data, we can find the proposed multiple line-intra prediction can reduce the MSE in most positions of a block. The benefit is especially obvious in the positions which are far away the reference line, namely the right-below region of a block. However, for the positions near the reference line, the gain of multiple line is very small, and there even exists a loss on the border (the red part in Fig. 6). This is because that the spatial correlation takes the most main influence if the spatial distance is very short. For this reason, the nearest reference line most likely can provide the best prediction, especially on the most-above row and the most-

Fig. 5. The residue compensation type for different intra prediction modes. (a) DC and planar, (b) mode 7 \(\sim\) 13, (c) mode 23 \(\sim\) 29, (d) mode 14 \(\sim\) 22, (e) mode 2, (f) mode 34.
left column of a block, and the utilize of further reference lines probably introduce the block discontinuities. From the lower layer data, we can see that the residue compensation can solve the problem of block discontinuities well when utilizing the further reference lines. The MSE has an obvious decrease on the border between the prediction block and the nearest reference line, indicated by the green color in Fig. 6.

C. Weighted Prediction

To further improve the prediction, we will continue weight the prediction generated by the nearest reference line $L_0$ after we perform the residue compensation. The joint utilization of further references and the nearest reference line can further suppress the influence of signal noise just like the principle that bi-prediction in inter coding, in where the predictions estimated from two reference pictures are weighted. In addition, the weighted prediction can provide a more reliable prediction when considering the strong spatial correlation of the nearest reference line. For this reason, we propose to weight the predictions from the two reference lines, and the weights are $[3/4, 1/4]$ respectively ($3/4$ is for the further reference line). To verify the effectiveness of weighted prediction, we calculate the MSE difference, namely the MSE of proposed method with weighted prediction minus the MSE of proposed method without weighted prediction. The MSE is still between the prediction block and original block. Fig. 7 shows the examples of $16 \times 16$ and $8 \times 8$ blocks under QP 27 and 32. From the figure, we can see that the MSE of most positions has a reduction if the weighted prediction is applied. For example, the average MSE difference of $16 \times 16$ block in BasketballDrill is $-15.153$ under QP 27, as shown in in the Fig. 7(c).

D. Fast Algorithms

As described in the previous section, the reference line index is with the CU level. In the proposed method, the best reference line is decided by the rate distortion (RD) cost, formulated by:

$$L_{\text{Best}} = \arg \min_{L_X} (D_{L_X} + \lambda \times R_{L_X})$$  \hspace{1cm} (7)$$

where $D_{L_X}$ is the distortion between the reconstructed block using reference line $L_X$ and the original block, $R$ is the corresponding bits, and $\lambda$ is the Lagrangian multiplier specified by HEVC. The encoder will check the reference lines from the nearest one $L_0$ to the farthest one, and choose the best one according to the RD cost, on CU level.

It means that the combinations which the encoder needs to traverse not only include the CU size, PU mode (two modes for intra prediction: $2N \times 2N$ and $N \times N$), and TU size but also including the reference line. This will approximately linearly increase the computational complexity with checking more reference lines. Considering the trade off between the compression efficiency and encoding complexity, we propose to use the near four reference lines, namely from $L_0$ to $L_3$. However, this paper also provides an alternative solution which is incorporated with several fast algorithms for some situations which are sensitive to the encoding complexity. And these acceleration algorithms are introduced subsequently.

Just as analyzed in section II, the reconstruction quality is an important motivation of the proposed method. In most cases, the reconstruction quality of the most boundary region is much worse than other regions as shown in Fig. 1 and is especially bad at block corners. To obtain the more detailed information, we define the block line (indicated by $BL_X$), as shown in Fig. 8(a). By analyzing large quantity of the data, we find another characteristic in many compressed pictures, which is the block line $BL_2$ does not have a obviously better quality than the block line $BL_1$. This characteristic is more common...
on large QP. The quality of $BL_2$ is even worse than that of $L_1$ in some cases. Fig. 8(b) ~ (d) shows some examples of the average variance of spatial quantization error in each block line. The average variance is calculated as:

$$AveV_{BL_X} = \frac{\sum_{i=0}^{N-1} v_{N-1-X,i} + \sum_{j=0}^{N-1} v_{j,N-1-X}}{2 \times N}$$

where the $v_{j,i}$ is the variance of the quantization error in location $(j,i)$, and $AveV_{BL_X}$ is the average variance of block line $BL_X$. From the examples shown in Fig. 8, we can see that the quality $BL_3$ has no advantage over that of $BL_1$ (Fig. 8 (d)). In Fig. 8 (b) and (c), the $AveV_{BL_3}$ is even larger than that of $BL_1$. However, in these examples, the quality of $BL_3$ is still better than that of $BL_1$. According to the structure of block line (Fig. 8 (a)) and reference line (Fig. 8), it is reasonable that using the quality of $BL_X$ to model the quality of $L_X$ because they contain the pixels with same location. Based on the aforementioned analysis on block line, we deduce that the quality of reference line $L_2$ has no obvious advantage over that of reference line $L_1$ in many cases. For this reason, we propose to skip the intra coding with $L_2$ in the fast solution. As we cannot deny the contribution that the reference line $L_2$ can reduce the influence of signal noise or occlusion by providing the potentially better prediction than $L_0$, we still keep the $L_2$ in the non-fast solution. It is noted that the disabling $L_2$ in the fast solution is a normative modification (i.e. there needs corresponding modification in the decoder) because we need modify the binarization of reference line index.

As the limitations of intra prediction, the larger prediction blocks (i.e. 64x64 and 32x32) generally introduce more prediction error. They are seldom selected under high bit rate, for small resolution sequences, or when there are a lot of textures. For this reason, the CU size of 64x64 is always not checked for further reference lines in the fast solution. If the neighbor PUs are with a small size, it means that current region is likely to be full with texture, and current block also probably should be encoded with small block for more elaborate prediction. For this reason, Lim et al. [19] propose to skip checking of $32 \times 32$ PU (i.e. $32 \times 32$ CU) when the above PU in above largest CU (LCU) and left PU in left LCU are both less than 16. In our method, we break the LCU constraint, the $32 \times 32$ CU is not checked for further reference lines when the above and left PU (they can locate in same LCU with current CU) are both less than 16.

As the diversity of content, the nearest reference line $L_0$ already predict the original block well sometimes. In this situation, the checking of further reference lines seems to be superfluous. So we propose to skip the encoding of further reference lines when the nearest reference line is shown as efficient enough. When we observe that:

$$C_{L_1} > f_1 \times C_{L_0}$$

where $C_{L_X}$ indicates the rate distortion cost of CU with reference line $L_X$ and the $f_1$ is a parameter set as 1.1 in the implementation, we will not check the reference lines which are further than the line $L_1$. In the encoder of HEVC reference software (HM), the intra mode $2N \times 2N$ will be checked in front of mode $N \times N$. If we find that the RD cost of mode $N \times N$ with the nearest reference line $L_0$ is larger than that of $2N \times 2N$ with the best reference line $L_{Best}$:

$$C_{N \times N \& L_0} > f_2 \times C_{2N \times 2N \& L_{best}}$$

, the further reference lines of mode $N \times N$ will not be examined because we deduce that the mode $2N \times 2N$ performs better than mode $N \times N$. Here $f_2$ is set as 1.2 in the implementation.

The whole procedure of intra coding in the encoder of HM can be divided into three main steps: coarse-grained direction decision (also known Rough Mode Decision, RMD) based on sum of absolute transformed differences cost, fine-grained direction decision based on RD cost with fixed TU size, and the check of best residual quadtree (RQT) size with the best direction. The second step will occupy the majority of computational complexity as several full encoding procedures will be performed. To accelerate the proposed scheme, the number of direction candidates (before that the number of most probable modes (MPM) is counted) will be cut by half in the second step for the further reference lines. In addition, the tool of rate distortion optimized quantization (RDOQ) is disabled in second step for further acceleration.

It should be noted that the all of the proposed fast algorithms only work for the further reference lines. The intra coding with the nearest reference line $L_0$ keeps unchanged. The flow char of the proposed fast algorithms is shown in Fig. 9.

IV. EXPERIMENTAL RESULTS

A. Experiment Setting

To verify the compression performance of the proposed multiple line-based intra prediction scheme, we implement it...
into the HEVC reference software HM-16.9 [20]. The test sequences include the whole range of HEVC standard test video sequences. They are arranged into five classes: Class A (4K), Class B (1080p), Class C (WVGA), Class D (WQVGA), Class E (720p) and Class F (synthetic).

The common test conditions specified in [21] is strictly followed in the test unless we point out otherwise explicitly. As our algorithms are designed for intra coding, we only test the all intra main configuration. The quantization parameters are set as 22, 27, 32, and 37. The results are evaluated by BD-Rate [22], where the negative number indicates bitrate saving and positive number indicates bitrate increasing.

### B. Experimental Results of Proposed Method

The experimental results of the proposed multiple line-based scheme with full search (i.e. disabling the proposed fast algorithms) are shown in Table I. The average bitrate saving is 2.2%, and the maximum bitrate saving is about 3.7% for BasketballDrill. As for computational complexity, we can see that the proposed scheme with full search will increase about 363% encoding time. This is because the encoder needs to iteratively check the intra coding with four reference lines. In addition, the post-processing of residue compensation and weighted prediction needs extra interpolations. Similar, as the existence of residue compensation and weighted prediction, the decoding time has a little increase.

In this paper, we also provide an alternative solution cooperated with several fast algorithms when the encoder is sensitive to the complexity. The experimental result of the proposed scheme with fast algorithms are shown in Table II. From the table, we can see that the increased encoding time has an obvious reduction, from 363% to 115%. Nevertheless, the coding efficiency only has a little drop. The coding gain is still up to 1.9%, in where the maximum bitrate saving is about 3.2% for BasketballDrill.

### C. More Analysis

From the results shown in Table I and Table II, we can see that the multiple-line (four-line) based intra prediction can significantly improve the coding efficiency. In the proposed method, the coding efficiency will improve if checking more reference lines in a range. More experimental results show that the coding gain continues to increase when the number of checked reference lines is from two to twelve. The Fig. 10 show the trend of coding gain with checking different number of reference lines. For convenience, only one frame is encoded. In the figure, the result of checking twelve lines is 2.82% on average bitrate saving. However, with the number of reference lines larger, the encoding complexity has a approximately

---

**Table I**

| Sequence       | BD-Bitrate Saving (%) |
|----------------|------------------------|
| Traffic        | −2.4%  | −1.6%  | −1.8%  |
| PeopleOnStreet | −2.6%  | −2.5%  | −2.4%  |
| Nebuta         | −2.3%  | −2.5%  | −1.9%  |
| SteamLocomotive| −1.4%  | −2.6%  | −2.8%  |
| Kimono         | −1.6%  | −1.4%  | −1.5%  |
| ParkScene      | −2.0%  | −1.2%  | −1.5%  |
| Cactus         | −2.5%  | −1.4%  | −2.2%  |
| BasketballDrive| −2.6%  | −2.5%  | −2.6%  |
| BQTerrace      | −2.7%  | −2.2%  | −2.5%  |
| BasketballDrill| −3.7%  | −4.0%  | −4.2%  |
| BQMall         | −2.0%  | −1.4%  | −1.6%  |
| PartyScene     | −2.0%  | −1.2%  | −1.2%  |
| RaceHorsesC    | −2.1%  | −1.6%  | −2.2%  |
| BasketballPass | −2.2%  | −1.9%  | −1.9%  |
| BQSquare       | −1.9%  | −1.1%  | −1.3%  |
| BlowingBubbles | −1.9%  | −1.2%  | −1.3%  |
| RaceHorses     | −2.1%  | −2.0%  | −2.2%  |
| FourPeople     | −2.1%  | −1.6%  | −1.6%  |
| Johnny         | −2.4%  | −3.1%  | −2.7%  |
| KristenAndSara | −2.1%  | −2.6%  | −2.4%  |
| BasketballDrillText | −3.2% | −3.2%  | −3.2%  |
| ChinaSpeed     | −1.8%  | −1.1%  | −1.2%  |
| SlideEditing   | −2.2%  | −1.3%  | −1.2%  |
| SlideShow      | −1.9%  | −1.2%  | −1.1%  |
| Average        | −2.2%  | −1.9%  | −2.0%  |

**Encoding / Decoding time**

|                | 463% | 115% |
|----------------|------|------|
TABLE II

| Sequence        | BD-Bitrate Saving (%) |          |          |
|-----------------|------------------------|----------|----------|
|                 | Y                      | U        | V        |
| Traffic         | -2.1%                  | -1.1%    | -1.2%    |
| PeopleOnStreet  | -2.3%                  | -1.7%    | -1.7%    |
| Nebuta          | -1.9%                  | -2.0%    | -1.5%    |
| SteamLocomotive | -1.1%                  | -0.9%    | -1.1%    |
| Kimono          | -1.4%                  | -0.9%    | -0.9%    |
| ParkScene       | -1.7%                  | -0.7%    | -1.0%    |
| Cactus          | -2.1%                  | -0.8%    | -1.3%    |
| BasketballDrive | -2.2%                  | -1.3%    | -1.4%    |
| BQTerrace       | -2.2%                  | -1.3%    | -1.4%    |
| BasketballDrill | -3.2%                  | -2.8%    | -3.0%    |
| BQMall          | -1.7%                  | -0.9%    | -1.0%    |
| PartyScene      | -1.5%                  | -0.7%    | -0.7%    |
| RaceHorsesC     | -1.8%                  | -1.2%    | -1.5%    |
| BasketballPass  | -1.9%                  | -1.1%    | -1.2%    |
| BQSquare        | -1.4%                  | -0.6%    | -0.8%    |
| BlowingBubbles  | -1.5%                  | -0.6%    | -0.8%    |
| RaceHorses      | -1.8%                  | -1.4%    | -1.4%    |
| FourPeople      | -1.9%                  | -1.1%    | -1.0%    |
| Johnny          | -2.0%                  | -1.9%    | -1.8%    |
| KristenAndSara  | -1.8%                  | -1.7%    | -1.4%    |
| BasketballDrillText | -2.8%              | -2.3%    | -2.3%    |
| ChinaSpeed      | -1.4%                  | -0.6%    | -0.7%    |
| SlideEditing    | -1.6%                  | -0.8%    | -0.7%    |
| SlideShow       | -1.5%                  | -0.7%    | -1.0%    |
| Average         | -1.9%                  | -1.29%   | -1.3%    |

Encoding / Decoding time: 215% / 114%

Fig. 10. The trend of coding gain with checking different number of reference lines. One Frame.

worse. For this reason, we skip checking the $L_2$ in the fast solution. In addition, we collect the reference line distributions in the actual bit stream to verify that the $L_2$ is less helpful when compared other reference lines. The Fig. 11 shows the ratio distribution of reference lines in the proposed method with full search. The sequence is Traffic. From the figure, we can see that the ratio of $L_2$ is much small than that of other reference lines on large QP. It follows similar trend for different CU sizes (from $32 \times 32$ to $8 \times 8$, we do not collect the information of $64 \times 64$ blocks as they are seldom chosen in the bit stream). The ratio difference between $L_2$ and other lines is smaller with QP smaller when comparing Fig. 11 (a) to (d). This is mainly that the quality difference of reference lines is smaller with QP smaller, and there is no quality difference for very small QP or lossless coding. From these results, we conclude that the $L_2$ is relatively less helpful on the whole, especially on large QP, and this also supports us to skip the checking $L_2$ in the fast solution. In addition, we find that the $L_0$ is still most chosen in the multiple line-based scheme as $L_0$ has the strongest spatial correlation.

V. Conclusion

This paper proposes a multiple line-based intra prediction scheme to improve the coding efficiency by utilizing the local further reference lines. A residue compensation procedure is introduced to suppress the discontinuities along the block border when using the further reference lines, followed by weighted prediction to further refine the prediction. Experimental results show the proposed algorithm improves the coding efficiency by 2.2% on average and up to 3.7%. In addition, this paper designs several acceleration algorithms when considering the encoding complexity. When enabling the fast algorithms, the encoding time only increase about 115%, but the coding gain is still up to 1.9% on average. The proposed method is compatible with many other solutions which only utilize single reference line, then can jointly optimize the intra prediction.
REFERENCES

[1] G. J. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, “Overview of the high efficiency video coding (hevc) standard,” Circuits and Systems for Video Technology, IEEE Transactions on, vol. 22, no. 12, pp. 1649–1668, 2012.

[2] J.-R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, “Comparison of the coding efficiency of video coding standards including high efficiency video coding (hevc),” Circuits and Systems for Video Technology, IEEE Transactions on, vol. 22, no. 12, pp. 1669–1684, 2012.

[3] J. Lainema, F. Bossen, W.-J. Han, J. Min, and K. Ugur, “Intra coding of the hevc standard,” Circuits and Systems for Video Technology, IEEE Transactions on, vol. 22, no. 12, pp. 1792–1801, 2012.

[4] L. Zhang, X. Zhao, S. Ma, Q. Wang, and W. Gao, “Novel intra prediction via position-dependent filtering,” Journal of Visual Communication and Image Representation, vol. 22, no. 8, pp. 687–696, 2011.

[5] F. Kamisli, “Intra prediction based on statistical modeling of images,” in Visual Communications and Image Processing (VCIP), 2012 IEEE. IEEE, 2012, pp. 1–6.

[6] ———, “Intra prediction based on markov process modeling of images,” Image Processing, IEEE Transactions on, vol. 22, no. 10, pp. 3916–3925, 2013.

[7] Y. Chen, J. Han, and K. Rose, “A recursive extrapolation approach to intra prediction in video coding,” in Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on. IEEE, 2013, pp. 1734–1738.

[8] S. Li, Y. Chen, J. Han, T. Nanjundaswamy, and K. Rose, “Rate-distortion optimization and adaptation of intra prediction filter parameters,” in Image Processing (ICIP), 2014 IEEE International Conference on. IEEE, 2014, pp. 3146–3150.

[9] D. Liu, X. Sun, F. Wu, S. Li, and Y.-Q. Zhang, “Image compression with edge-based inpainting,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 17, no. 10, p. 1273, 2007.

[10] D. Liu, X. Sun, F. Wu, and Y.-Q. Zhang, “Edge-oriented uniform intra prediction,” Image Processing, IEEE Transactions on, vol. 17, no. 10, pp. 1827–1836, 2008.

[11] D. Doshkov, P. Ndjiki-Nya, H. Lakshman, M. Köppel, and T. Wiegand, “Towards efficient intra prediction based on image inpainting methods,” in Picture Coding Symposium (PCS), 2010. IEEE, 2010, pp. 470–473.

[12] Y. Zhang and Y. Lin, “Improving hevc intra prediction with pde-based inpainting,” in Asia-Pacific Signal and Information Processing Association, 2014 Annual Summit and Conference (APSIPA). IEEE, 2014, pp. 1–5.

[13] X. Qi, T. Zhang, F. Ye, A. Men, and B. Yang, “Intra prediction with enhanced inpainting method and vector predictor for hevc,” in Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on. IEEE, 2012, pp. 1217–1220.

[14] Y.-H. Lai and Y. Lin, “Error diffused intra prediction for hevc,” in Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on. IEEE, 2015, pp. 1424–1427.

[15] C. Chen, B. Zeng, S. Zhu, Z. Miao, and L. Zeng, “A new block-based coding method for hevc intra coding,” in Multimedia & Expo Workshops (ICMEW), 2015 IEEE International Conference on. IEEE, 2015, pp. 1–6.

[16] T. K. Tan, C. S. Boon, and Y. Suzuki, “Intra prediction by template matching,” in Image Processing, 2006 IEEE International Conference on. IEEE, 2006, pp. 1693–1696.

[17] M. A. Robertson and R. L. Stevenson, “Det quantization noise in compressed images,” Circuits and Systems for Video Technology, IEEE Transactions on, vol. 15, no. 1, pp. 27–38, 2005.

[18] G. J. S. J.-R. O. B. Bross, W.-J. Han and T. Wiegand, High Efficiency Video Coding (HEVC) Text Specification Draft 7, JCTVC-I1003, 9th Meeting: Geneva, May 2012.

[19] K. Lim, J. Lee, S. Kim, and S. Lee, “Fast pu skip and split termination algorithm for hevc intra prediction,” Circuits and Systems for Video Technology, IEEE Transactions on, vol. 25, no. 8, pp. 1335–1346, 2015.

[20] https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.9/.

[21] F. Bossen, Common test conditions and software reference configurations, JCTVC-L1100, 12th Meeting: Geneva, Jan 2013.

[22] G. Bjntegaard, Improvements of the BD-PSNR model, Document VCEG-AI11, July 2008.