# Improving Intra Prediction in High-Efficiency Video Coding

Haoming Chen, Tao Zhang, Ming-Ting Sun, *Fellow, IEEE*, Ankur Saxena, *Member, IEEE*, and Madhukar Budagavi, *Senior Member, IEEE*

***Abstract*— Intra prediction is an important tool in intra-frame video coding to reduce the spatial redundancy. In current coding standard H.265/high-efficiency video coding (HEVC), a copying-based method based on the boundary (or interpolated boundary) reference pixels is used to predict each pixel in the coding block to remove the spatial redundancy. We find that the conventional copying-based method can be further improved in two cases: 1) the boundary has an inhomogeneous region and 2) the predicted pixel is far away from the boundary that the correlation between the predicted pixel and the reference pixels is relatively weak. This paper performs a theoretical analysis of the optimal weights based on a first-order Gaussian Markov model and the effects when the pixel values deviate from the model and the predicted pixel is far away from the reference pixels. It also proposes a novel intra prediction scheme based on the analysis that smoothing the copying-based prediction can derive a better prediction block. Both the theoretical analysis and the experimental results show the effectiveness of the proposed intra prediction method. An average gain of 2.3% on all intra coding can be achieved with the HEVC reference software.**

***Index Terms*— Intra prediction, coding gain, high efficiency video coding.**

## I. Introduction

INTRA frame coding is essential in both still image and video coding. In the block-based coding scheme, the spatial redundancy can be removed by utilizing the correlation between the current pixel and its neighboring reconstructed pixels. From the differential pulse code modulation (DPCM) in the early video coding standard H.261 to the angular intra prediction in the latest H.265/HEVC [1], different intra prediction schemes are employed. Almost without exception, linear filters are used in these prediction schemes. In H.264/AVC and H.265/HEVC, a copying-based intra prediction is used. In this scheme, a reference pixel (or a filtered one over its neighboring pixels) is copied as the predictor of current pixel.[1] Recently, recursive prediction methods with 3-tap filters [2] or 4-tap filters [3] are proposed to improve the intra prediction accuracy. In both methods, the prediction weights are derived under the first-order 2D Gaussian Markov model, in which the parameters are trained from real video sequences. In [4], it compares the copying-based intra prediction used in H.264/AVC and the recursive intra prediction, and presents a new training method for the parameters in the first-order 2D Gaussian Markov model. Fundamentally, the design of linear intra prediction can be formulated as finding suitable prediction weights. It has been shown that given the statistics of the coding blocks as well as the reference pixels, the Wiener filter [5] is the optimum filter leading to the minimum prediction mean square error (MSE) [6]. Based on the statistical model trained from real images, a position-dependent filtering intra prediction is proposed in [7]. Each position under different block size and intra prediction mode has its own (Wiener) filtering weights. However, a large table of filtering weights is needed, and it is unsuitable for HEVC with block sizes up to $64 \times 64$ and the number of modes up to 35. A partial-differential-equation-based (or briefly PDE-based) image inpainting technique is applied in the intra prediction [8]. It assumes the image patch is smooth so that the prediction block is generated by solving a PDE problem with known boundary pixels. An improved method using mode-dependent angular masks is proposed in [9]. However, the computation complexity for solving the PDE problem is much higher than the linear intra prediction scheme. In previous linear intra prediction methods, the boundary reference pixels are assumed to have high quality and follow the 2D Gaussian Markov model. However, in the real sequences, some boundary pixels may not be available, or have noise and/or quantization errors [10], [11].

In this paper, we first theoretically analyze the optimal prediction assuming the pixels in the video frames follow an underline first-order Gaussian Markov model. It shows that the copying-based method is a good approximation of the optimal prediction when the inter-pixel correlation is high. However, when the pixel is far away from the reference, the correlation is low, that may need a smaller prediction weight. We then define the pixels with error as those pixels with pixel values

Manuscript received July 23, 2015; revised January 1, 2016 and March 11, 2016; accepted May 10, 2016. Date of publication May 26, 2016; date of current version June 22, 2016. This work was supported by Samsung Research America. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Wen Gao.

H. Chen and M.-T. Sun are with the Department of Electrical Engineering, University of Washington, Seattle, WA 98105 USA (e-mail: eehmchen@ee.washington.edu; sun@ee.washington.edu).

T. Zhang was with the University of Washington, Seattle, WA 98105 USA. He is now with the Harbin Institute of Technology, Harbin 150001, China (e-mail: taozhang.hit@hotmail.com).

A. Saxena is with Nvidia, Santa Clara, CA 95050 USA (e-mail: saxena.ankur0@gmail.com).

M. Budagavi is with Samsung Research America, Richardson, TX 75082 USA (e-mail: m.budagavi@samsung.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2016.2573585

[1]To avoid confusion, it is noted that the "copying-based intra prediction" is unrelated with the newly adopted "intra block copy" mode in the screen content video coding extension (HEVC-SCC).

deviate from the underline model. This error will account for the quantization errors, noises, and edge pixels. It shows that the availability and the errors of the boundary pixels affect the intra prediction accuracy. The theoretical coding gain of different intra prediction methods are compared. Based on the theoretical analysis, the paper presents a novel intra prediction scheme via properly smoothing the copying-based intra prediction.

It should be noted that besides using one-pixel width reference pixels adjacent to the current block, some other intra prediction schemes, e.g., intra block copy [12]–[14] and palette coding [15], are proposed in the screen content coding [16]. Although these methods show a very significant gain in the screen content coding, the gain on the natural content video coding is not as good as that of the conventional linear prediction method. Hence, in this paper, we focus on the improvements on the linear intra prediction scheme using boundary reference pixels.

The reminder of this paper is as follows: Section II discusses the optimal intra prediction when predicted pixels are far away from reference pixels and the reference pixels have deviation. A theoretical coding gain is proposed as well. Section III proposes the novel intra prediction scheme with block-size-dependent iterative filtering and shows the design of this filters. The experimental results are presented in Section IV. Section V concludes this paper.

## II. Theoretical Coding Gain With Intra Prediction

The theoretical coding gain comparison is used in the design of transform coding [17]–[19]. This section utilizes the coding gain to discuss different prediction schemes and the effect when the predicted samples are far away from the reference sample and reference samples have deviation.

### A. Intra Prediction in 1D Source Without Error

Consider a 1D sequence, $X = [x_0, x_1, \ldots, x_N]^T$, following the zero-mean unit-variance first-order Gaussian Markov chain model, its auto-correlation matrix is:

$$E[XX^T] = \begin{pmatrix} 1 & \rho & \rho^2 & \cdots & \rho^N \\ \rho & 1 & \rho & \cdots & \rho^{N-1} \\ \rho^2 & \rho & 1 & \cdots & \rho^{N-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho^N & \rho^{N-1} & \rho^{N-2} & \cdots & 1 \end{pmatrix}, \tag{1}$$

where the $\rho$ is the correlation coefficient, which is usually close to 1.

Suppose the first sample $x_0$ is the boundary value which is used to predict $x_{\mathrm{i}}$, $i = 1, ..N$, there exists a prediction matrix $P$ generating the predictors $\hat{X}$ as follows:

$$P = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 \\ p_1 & 0 & 0 & \cdots & 0 \\ p_2 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p_N & 0 & 0 & \cdots & 0 \end{pmatrix}, \quad \text{and} \tag{2}$$

$$\hat{X} = PX, \tag{3}$$

where $p_i$, $i = 1, \ldots, N$, are the prediction weights on $x_0$ for each samples. Note that only the first column of $P$ has non-zero values, which means the prediction only depends on $x_0$. The problem of finding the optimum prediction $P_{\text{opt}}$ leading to the least predictor error can be formulated as

$$P_{\text{opt}} = \arg\min_P E \sum_{i=1}^{N} (x_i - p_i x_0)^2 \tag{4}$$

It is easy to get the optimum $p_i$ by taking the derivatives of Eq. (4) with respect to $p_i$ and set it to zero. The solution is

$$p_{i,\text{opt}} = E(x_i x_0)/E(x_0 x_0). \tag{5}$$

Substitute the auto-correlation with the values in Eq. (1), the optimum prediction weights under the first-order Gaussian Markov chain model is

$$p_{i,\text{opt}} = \rho^i, \quad i = 1, \ldots, N. \tag{6}$$

With optimum prediction matrix, the residuals $Y = [y_0, y_1, y_2, \ldots, y_N]^T$, is derived by $X - PX$

$$Y = (I - P)X, \tag{7}$$

where the $I$ is the identity matrix. The element in the auto-correlation matrix of $Y$ is

$$E[y_i y_j] = \rho^{|i-j|} - \rho^{i+j}, \quad i, j = 1, \ldots, N \tag{8}$$

The intra prediction process in Eq. (7) can be treated as a transform and the coding gain of a transform is the ratio of the geometric mean of the transform domain sample variance before and after the transform [20]. In the intra prediction case, since the reference samples (e.g., $x_0$ in the 1D case) are unchanged, the variance of these samples is excluded in the coding gain calculation. The coding gain defined in decibel is

$$G = 10\log_{10}\left(\left(\prod_{i=1}^{N}\sigma_{x_i}^2\right)^{1/N} \Big/ \left(\prod_{i=1}^{N}\sigma_{y_i}^2\right)^{1/N}\right), \tag{9}$$

where the variances of $x_i$ and $y_i$ are the auto-correlation $E(x_{\mathrm{i}}x_{\mathrm{i}})$ and $E(y_i y_i)$ since they are zero-mean. Substitute the variances with values in Eq. (1) and Eq. (8), the coding gain with optimal prediction weights in 1D first-order Gaussian Markov source is

$$G_{1D,opt} = -\frac{10}{N}\sum_{i=1}^{N}\log_{10}(1 - \rho^{2i}) \tag{10}$$

The coding gain of the copying-based intra prediction can be derived with Eq. (9) similarly. Note that when $N$ is small, copying the $x_0$, i.e., $p_i = 1$ in Eq. (2), is a good approximation of the optimum intra prediction. However, when $N$ is larger, the copying-based method is inefficient. Eq. (6) shows that when a sample has large index (far away from $x_0$), the prediction weight should be small. To illustrate the coding gain difference between the optimal prediction weights and the copying-based method when $N$ changes, one example is given as follows: a 1D first-order Gaussian Markov chain $[x_0 x_1 x_2 \ldots x_N]$ with $\rho = 0.95$. For different $N$ from 4 to 32, the coding gains with optimal prediction weights derived in Eq. (6) and the copying-based intra prediction are shown

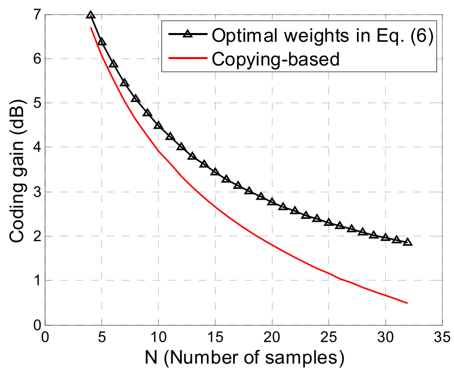



Fig. 1. Coding comparison under different sizes of samples. For larger size samples, the copying-based method is much less efficient than the optimal prediction weights.

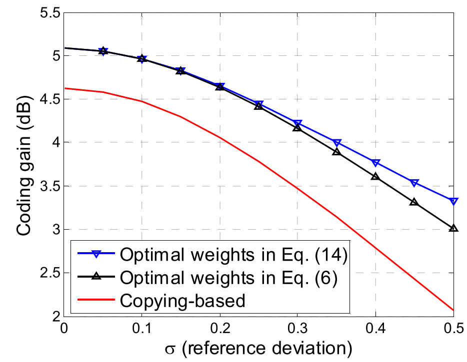



Fig. 2. Coding gain comparison under different levels of reference deviation. When the deviation is large, the optimal weights derived in Eq. (14) considering the deviation is much better than the weights derived in Eq. (6), which does not considering the deviation.

in Fig. 1. It is clear that the copying-based prediction is always less efficient than the optimal prediction. Moreover, the coding gain difference between these methods is larger when $N$ is increasing.

### *B. Intra Prediction for 1D Source With Error*

In the previous 1D case, it shows that when the predicted samples are far away from the reference sample. The copying-based intra prediction is not a good approximation of the optimal intra prediction. Note that, in this case, the reference sample has no error. However, in a real sequence, the reference sample is always a reconstructed one, with quantization error. Hence, the reference sample is

$$\hat{x}_0 = x_0 + \delta, \tag{11}$$

where the $\delta$ is the error on the reference pixels. The $\delta$ is zero-mean, independent to all other samples and has a variance of $\sigma^2$. It has been pointed out in [10] that $\delta$ can cover the case that the reference sample is unavailable (in practical video coding, the unavailable reference pixels are always filled with neighboring known pixels) or when there is noises on the original reference samples.

With the errors in the reference samples, the sequence and its auto-correlation matrix is rewritten as

$$\hat{X} = \begin{bmatrix} \hat{x}_0 & x_1 & \ldots & x_N \end{bmatrix}^T, \tag{12}$$

$$E[\hat{X}\hat{X}^T] = \begin{pmatrix} 1+\sigma^2 & \rho & \rho^2 & \cdots & \rho^N \\ \rho & 1 & \rho & \cdots & \rho^{N-1} \\ \rho^2 & \rho & 1 & \cdots & \rho^{N-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho^N & \rho^{N-1} & \rho^{N-2} & \cdots & 1 \end{pmatrix}. \tag{13}$$

Note that the only difference between Eq. (5) and Eq. (13) is the auto-correlation of the reference sample. The optimum prediction weights in the prediction matrix $P$ of Eq. (2) are:

$$p_{\mathrm{i}} = \rho^i/(1+\sigma^2), \quad i = 1, \ldots, N. \tag{14}$$

Compare Eq. (14) with Eq. (6), it shows that a smaller intra prediction weights are desired in case when the reference samples have error. For a relatively large error $\delta$, the optimum weights derived by Eq. (6) is sub-optimum. To illustrate the effects of this reference deviation, one example is shown as follow. For a source with $\rho = 0.95$, $N = 8$, and $\sigma = 0 \sim 0.5$, the coding gain with three intra prediction methods 1) derived by Eq. (14), 2) derived by Eq. (6), and 3) copying-based prediction are shown in Fig. 2. The figure shows that when the deviation associated with the reference samples increases, the intra prediction weights considering the error is much better than weights without considering the deviation and the copying-based method.

### *C. Intra Prediction for 2D Directional Sources Without Error*

The previous analysis on the 1D case can be extended to the 2D case. Denote $x_{i,j}$, $(i, j = 0, \ldots, N)$ as the a pixel on the $i$-th row and $j$-th column in an image. Under the 2D first-order Gaussian Markov model with zero mean and unit variance, an isophote auto-correlation model of two pixels $x_{i,j}$ and $x_{p,q}$ is used as follows:

$$E[x_{i,j}x_{p,q}] = \rho^{\sqrt{|i-p|^2+|j-q|^2}} \tag{15}$$

where $\rho$ is the correlation coefficient.

Considering the image source has a dominating direction, the correlation model can be extended to [17], [21]

$$E[x_{i,j}x_{p,q}] = \rho^{\sqrt{d_1^2(\alpha)+\eta^2 d_2^2(\alpha)}}, \tag{16}$$

where,

$$\begin{cases} d_1(\alpha) = (j-q)\cos\alpha - (i-p)\sin\alpha \\ d_2(\alpha) = (j-q)\sin\alpha + (i-p)\cos\alpha. \end{cases} \tag{17}$$

In this model, $\alpha$ is the dominating direction, e.g., $\alpha = 0$ means the source has a vertical direction. $d_1$ and $d_2$ are the distance between $x_{i,j}$ and $x_{p,q}$ along the dominating direction and perpendicular to that direction, respectively. $\eta$ ($\geq$1) is the parameter indicating the directionality. A larger $\eta$ means the correlation along the dominating direction is higher than the perpendicular direction. A typical set of parameters $\rho = 0.99$ and $\eta = 5$ (1 for DC and Planar modes) are used. These parameters are trained on the real video sequences and the details are described in the Appendix A.

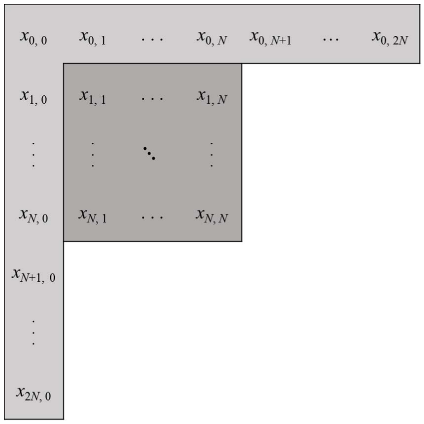



Fig. 3. Current coding block and its neighboring reference pixels.

For a 2D source, the coding block and its neighboring reference pixels can be concatenated into a tall vector. As in the HEVC, for a $N \times N$ block, the $2 \times N$ pixels above and above-right, and $2 \times N$ pixels left and left-bottom to the blocks, as well as the top-left one pixel, are used as the reference pixels, as shown in Fig. 3.

Here it assumes all reference pixels are available (or filled with some methods). So there is a total of $4 \times N + 1 + N^2$ samples in the vector:

$$X = [x_{0,0}, x_{1,0}, \ldots, x_{2N,0}, x_{0,1}, \ldots, x_{0,2N}, x_{1,1}, x_{2,1}, \ldots, x_{N,N}]^T \quad (18)$$

The first $4 \times N + 1$ samples are the references, followed by $N^2$ samples in the current coding block, which is concatenated column-wise. For simplicity, the 1D vector $X$ is rewritten as

$$X = [A^T \quad B^T]^T \quad (19)$$

where $A = \{a_i\}$, $i = 1, \ldots, 4N+1$, are the reference samples and $B = \{b_j\}$, $j = 1, \ldots, N^2$ are the coding block samples.

The intra prediction can also be formulated as a matrix multiplication, similar to Eq. (2) an Eq. (3). The prediction matrix $P$ is

$$P = \begin{pmatrix} \mathbf{0}_{(4N+1)\times(4N+1)} & \mathbf{0}_{(4N+1)\times N^2} \\ W_{N^2\times(4N+1)} & \mathbf{0}_{N^2\times N^2} \end{pmatrix} \quad (20)$$

where $W = \{w_{i,j}\}$ ($i = 1, \ldots, N^2$, $j = 1, \ldots, 4N+1$) is the matrix of weights. Each row of matrix is a set of prediction weights on all reference pixels.

Similar to the 1D source, the residual is derived by Eq. (7) and the coding gain can be calculated via Eq. (10). The problem of finding the optimum prediction weights is formulated as

$$W_{\text{opt}} = \arg\min_{W} E\left[\|B - W \bullet A\|_2^2\right]. \quad (21)$$

Let $w_i$ be the $i$-th row of $W$, the problem in Eq. (21) is equivalent to solving each row of $W$,

$$w_{i,opt} = \arg\min_{w_i} E\left[\|b_i - w_i \bullet A\|_2^2\right], \quad i = 1, .., 2N. \quad (22)$$

The solution is

$$w_{i,opt} = E(AA^T)^{-1} \bullet E(b_i A), \quad (23)$$

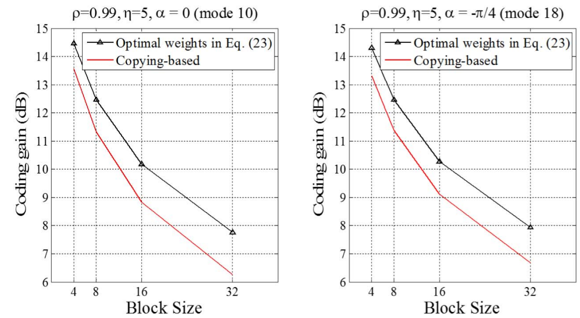



Fig. 4. Coding gain comparison between optimal weights intra prediction and the copying-based method under two directional models (left: horizontal, right: diagonal-right-down).

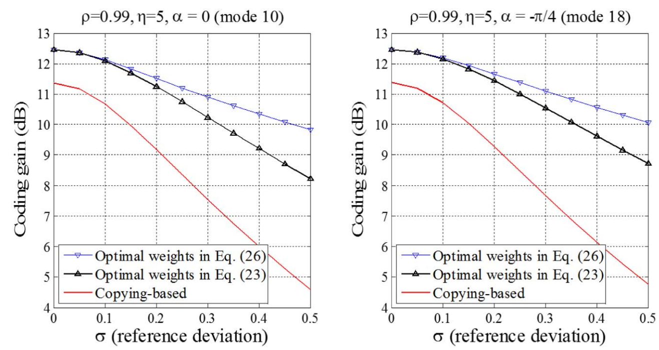



Fig. 5. Coding gain comparison under different levels of reference deviations. When the deviation is large, the optimal weights derived in Eq. (26) considering the deviation is much better than the weights derived in Eq. (23) without considering the deviation.

where $E(AA^T)$ is the auto-correlation matrix of reference samples and $E(b_i A)$ is the correlation between the sample $b_i$ and all reference samples. Actually, Eq. (22) is the Weiner filter.

Similar to 1D case, for a large block, the pixels far away from the reference boundary has lower correlation. One example as follow shows the coding gain difference between the copying-based intra prediction and the optimal weights intra prediction. In this example, the 2D samples follow the model in Eq. (16) with $\rho = 0.99$, $\eta = 5$. Two directions $\alpha = 0$ and $\alpha = -\pi/4$ are tested, which means the block has a horizontal direction or a diagonal down right direction. For $\alpha = 0$, the copying-based method follows the intra prediction mode 10, and for $\alpha = -\pi/4$, the copying-based method follows the intra prediction mode 18. The block size varies from 4 to 32. The coding difference under these two directions are shown in Fig. 4.

From above analysis, it is similar to the 1D case that in a large block, when there are more samples far away from the reference samples, the copying-based prediction is less efficient.

### *D. Intra Prediction on 2D Directional Sources With Error*

It has been shown in Section II-B that in 1D source, the error on reference samples affects the optimum intra prediction weights. The effects on 2D sources are analyzed as follows.

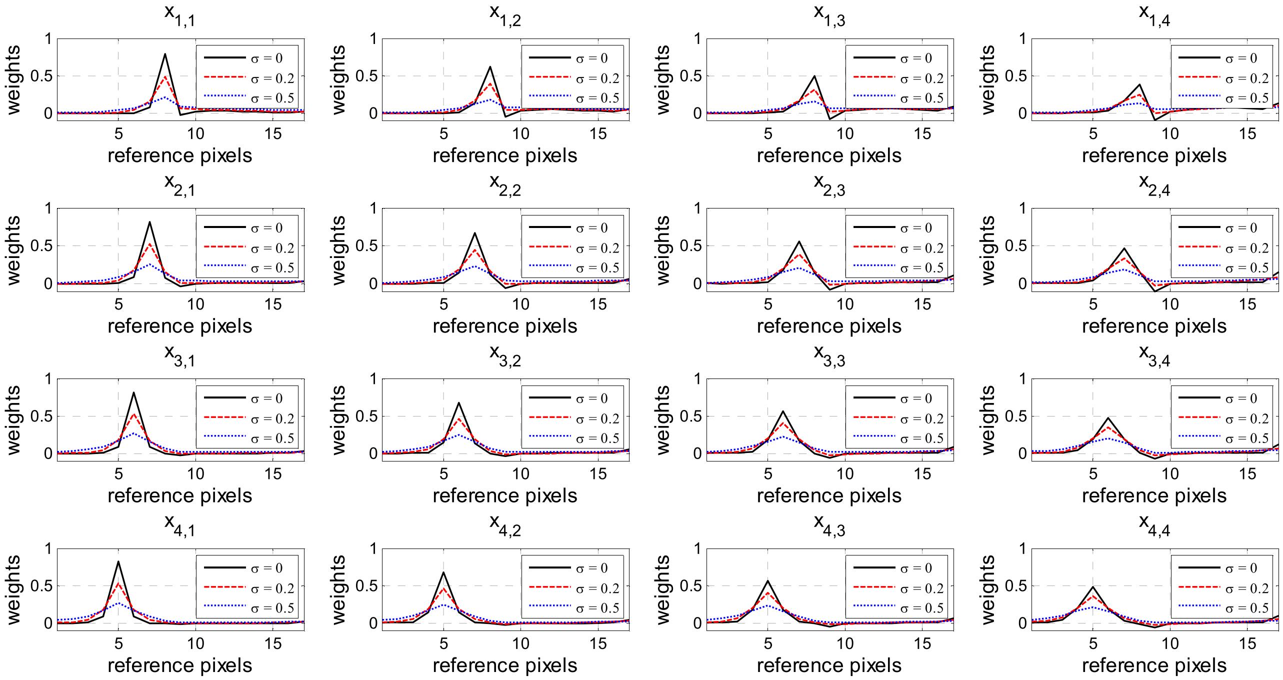



Fig. 6. These 16 figures show the prediction weights on all $4 \times 4$ pixels. In each figure, three lines present the weights of each reference pixels under different error level (solid line: $\sigma = 0$, dashed line: $\sigma = 0.2$ and dotted line: $\sigma = 0.5$). The index of the reference pixels correspond to the position of 17 reference pixels in the clock-wise direction, for the bottom-left $x_{8,0}$ to the top-right $x_{0,8}$.

Denote the distorted reference samples as

$$\bar{x}_{i,j} = x_{i,j} + \delta_{i,j}, \tag{24}$$

where $\delta_{i,j}$ is zero-mean, independent to all the samples and has a variance of $\sigma^2$. Hence, the Eq. (19) becomes

$$X = \left[ \bar{A}^T \quad B^T \right]^T, \tag{25}$$

where $\bar{A}$ is the vector of deviated reference samples. The solution in Eq. (22) is updated with

$$w_{i,opt} = E(\bar{A}\bar{A}^T)^{-1} \bullet E(b_i \bar{A}). \tag{26}$$

Compare Eq. (26) to Eq. (23), note that $\delta_{i,j}$ is independent to $b_i$, the change of optimal prediction weights only comes from the inverse of new auto-correlation matrix. The effects of reference deviation on the coding gain are shown in Fig. 5. In these figures, two different directional models (horizontal and diagonal-down-right) are tested. The $\sigma$ is from 0 to 0.5. It can be seen that when the deviation is relatively large, the optimal weights in Eq. (26) considering the deviation is much better than the weights by Eq. (23) which does not consider the deviation.

From the previous analysis, we have the conclusion that the current copying-based method in intra prediction is less efficient in two cases: 1) when more predicted samples are far away from the reference sample and 2) the reference samples have relatively larger deviations. One straight-forward idea is to apply the optimal weights instead of copying-based method in these two cases. Since all the correlation can be derived via the 2D source model in Eq. (16), or by training real image data [7], an optimum prediction weights table can be calculated. However, the weights are position-dependent, the size of weights table would be $N^2 \times (4N+1)$. In HEVC, the $N$ could be 4, 8, 16 and 32, hence there are 151120 numbers in a weights table for one directional model. Consider there are 34 intra prediction directions, each one having a different correlation matrix, there are more than $4 \times 10^6$ numbers! Hence, the position-dependent weights are not feasible for the HEVC intra prediction scheme. Hence, we like to find an approximation to the optimal weights.

From Eq. (6) and Eq. (14) derived in the 1D case, it is clear that a smaller weights would be better for aforementioned two cases where the copying-based method is inefficient. However, the derivation of the analytical solution to the optimum weights in 2D case is too complicated. A quantitative simulation is shown to compare the weights. In this simulation, a $4 \times 4$ block is predicted by its 17 neighboring references samples as shown in Fig. 3. Each sample is zero-mean and unit-variance. The correlation among all 33 samples is modeled by Eq. (16) with $\rho = 0.99$, $\eta = 5$ and $\alpha = 0$, which is the horizontal direction. For all samples in the block, the optimal weights can be derived, given the error $\delta$ (with variance $\sigma$). Different $\sigma = 0$, 0.2 and 0.5 are simulated and the corresponding weights are shown in Fig. 6.

The following observations can be derived from Fig. 6.

1) Under the same level of deviation, the prediction weights on the pixels far away from the reference along the prediction direction are smaller. In this example, the prediction direction is horizontal. From left to the right, the optimal weights under the same deviation level are decreasing.
2) On the same pixel, when the deviation level increases, the weights are also decreased. It can be proved that when the error is very large, the weights will approach to the same value, which is shown in the Appendix B.

**Algorithm 1** Iterative Filtering Intra Prediction

Input: An $N \times N$ coding block with its neighboring $4N$+1 reference pixels, smoothing filter $K$ and filtering iteration numbers $T$.

Output: An $N \times N$ prediction block *Pred*

1. Generate an intra predictor block $Blk_0$ using the standard copying-based H.265/HEVC angular intra prediction method as an initial block.
2. Convolve the $Blk_0$ with filter kernel $K$ recursively $T$ times.
   **for** $i = 1 : T$ **do**
   $Blk_i$, = conv($Blk_{i\text{-}1}$, $K$)
   **end for**
   The conv() function means the $Blk_{i\text{-}1}$ is convolved with kernel $K$. Note that although the left and upper reference boundary pixels are used in the convolution to determine the center pixel value, their values are not changed.
3. $Pred = Blk_T$

Note that in the above example, the pixels are assumed to be zero-mean, so that the sum of weights can be less than 1, which does not cause DC drift. However, in the real image, the pixel value is among 0 to 255, the DC value should be subtracted from the reference samples so that they are zero-mean. And then a weighted prediction are applied on these sample and the DC value is added to keep the samples from DC drifting. If the DC value is estimated by using the average of some neighboring pixels, it can be seen that a smaller prediction weight for zero-mean sources leads to a smoother weighted prediction on neighboring references. In the next section, we propose to use an iterative filtering method to approximate the smoothing weighted prediction.

## III. Proposed Rate-Distortion Optimized Iterative Filtering Intra Prediction

The example in the previous section shows that smoothed prediction weights instead of copying-based method is more suitable for pixels far away from the reference samples along the direction prediction and the reference samples have deviation. This section proposes an iterative filtering method as an extra intra prediction mode in addition to the copying-based method to handle this situation.

### A. Iterative Filtering Intra Prediction

To smooth the prediction weights, a copying-based predictor block is first generated, then the predictor block is convolved with a smoothing filter. Since the larger block has more pixels far away from the boundary, a stronger smoothing is needed. To have a smoother predictor, a larger smoothing filter is needed or the smoothing filter is applied multiple times. In this scheme, the latter one is chosen since it is easier to choose the filtering times (discussed later) other than to design the filter parameters in a large filter. The iterative filtering algorithm is summarized as follows in Algorithm 1.

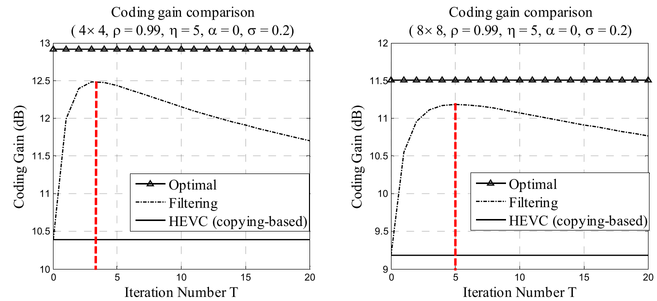



Fig. 7. Coding comparison between 1) the optimal intra prediction; 2) the iterative filtered intra prediction and 3) the HEVC (copying-based) intra prediction mode 10. The best performance of iterative filtering intra prediction for $4 \times 4$ and $8 \times 8$ blocks appear at iteration time $T = 3$ and 5, respectively.

### B. Filter Design and Iteration Number Selection

Given a smoothing filter, the proper iteration time needs to be determined. In this part, the coding gain defined in Eq. (9) is used to evaluate the best iteration number. According to our simulation, the results are not sensitive to the selection of filter kernel. It is pointed out in [9] that a mode dependent kernel can be used. However, according to our tests, mode-dependent kernels do not improve the performance of our method. It is also possible to apply multiple filters and select the best one by RD optimization, but the extra overhead to indicate the kernel will increase the bitrate and the computing complexity. Hence, the following smoothing filter is used:

$$K = \frac{1}{6}\begin{pmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{pmatrix} \tag{27}$$

Consider an $N \times N$ block with $4N + 1$ neighboring references pixels. Each pixel is zero-mean and unit-variance. The correlation among all pixels is modeled by Eq. (16) with $\rho = 0.99$, $\eta = 5$ and $\alpha = 0$, which is the horizontal direction. The deviation with $\sigma = 0.2$ is added on the reference pixels. The following 3 intra prediction schemes are compared: 1) optimal intra prediction derived by Eq. (26); 2) Iterative filtering intra prediction via the filter $K$ in Eq. (27) with iteration time of $T = 1, 2, \ldots, 20$; and 3) the copying-based intra prediction in the HEVC with mode 10 (horizontal); The coding gain comparison under different $T$ is showed in Fig. 7.

From the above example, it shows that the optimal weights scheme outperforms both copying-based and the proposed iterative filtering intra prediction schemes. However, as mentioned in the previous section, the optimal weights intra prediction needs a large weights table, which is infeasible in current HEVC standard. For the iterative filtering scheme on $4 \times 4$ and $8 \times 8$ blocks, the coding gain increases as the iteration grows and then decreases after the best iteration number. The best iteration number of 3 and 5 achieve the highest gain in $4 \times 4$ and $8 \times 8$ blocks, respectively. Hence, in the design of the proposed intra prediction, the iteration number in the horizontal mode is set to 3 for $4 \times 4$ and 5 for $8 \times 8$.

The similar simulation is done on the DC mode and other HEVC angular modes, with different size include $4 \times 4$, $8 \times 8$, $16 \times 16$ and $32 \times 32$. The simulation results for DC mode is shown in Fig. 8.

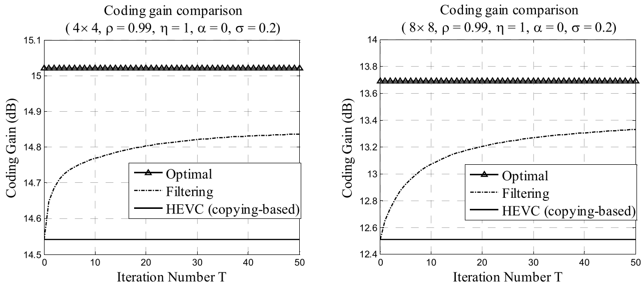



Fig. 8. Coding comparison between the optimal, the HEVC mode 1 (DC) and the iterative filtered intra prediction schemes on a 2D source without dominating direction ($\eta = 1$, $\rho = 0.99$ and $\sigma = 0.2$). The coding of iterative filtering intra prediction will converge after about $T = 30$.

TABLE I
BEST ITERATION NUMBERS UNDER DIFFERENT MODES AND BLOCK SIZES

| Modes | 4×4 | 8×8 | 16×16 | 32×32 |
|---|---|---|---|---|
| 0 and 1 (Planar and DC) | 32 | 32 | 32 | 32 |
| 2 ~ 34 (Angular Modes) | 3 | 5 | 10 | 25 |

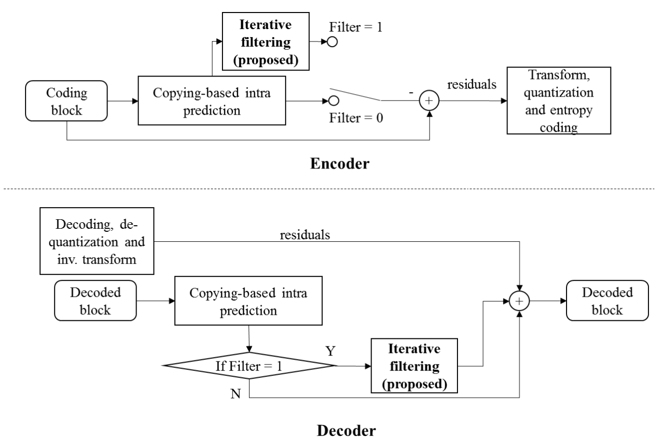



Fig. 9. The flowchart of the proposed intra prediction scheme on both encoder and decoder side. The iterative filtering intra prediction is added as an extra coding mode.

The best iteration numbers under different situations are listed in Table I.

## C. Rate-Distortion-Based Mode Selection

The proposed iterative filtering method is used as an extra mode to the existing copying-based method in the intra prediction process. The encoder and decoder flow-chart is shown in Fig. 9.

The Filter flag in Fig. 9 is selected based on the rate-distortion cost of each Coding Unit (CU) and encoded into the bit stream by the entropy encoder. Note that in the current HEVC standard, the intra prediction is applied on the Transform Unit (TU) level, which means the CU can be further split into a quad-tree structure and then the intra prediction is applied on the leaves of this quad-tree, which is called residual quad-tree (RQT). RQT can enhance the prediction accuracy since the reference is closer to the predicted block. To achieve the best performance, the proposed iterative filtering should be tested on each TU and different TUs may choose different modes (with or without filtering). However, this increases overhead bits and the searching time significantly. Therefore, in the proposed scheme, the RDO-based selection is tested on the CU level and only one Filter flag is encoded for each CU.

TABLE II
BD-BITRATE REDUCTION WITH AI, RA, LDB AND LDP CONFIGURATION ON FULL SEQUENCES

| | | AI | RA | LDB | LDP |
|---|---|---|---|---|---|
| 2K (1920×1080) | Kimono | -2.7% | -1.0% | -0.2% | -0.7% |
| | ParkScene | -1.6% | -1.0% | -0.5% | -0.6% |
| | Cactus | -1.5% | -0.8% | -0.4% | -0.7% |
| | BasketballDrive | -1.7% | -1.0% | -0.5% | -1.0% |
| | BQTerrace | -0.6% | -0.3% | -0.2% | -0.1% |
| 4K (3840×2160, except hotel 4096×2048) | Chuno_s4 | -2.3% | -1.3% | -0.9% | -1.4% |
| | Chuno_s31 | -3.2% | -2.0% | -2.1% | -2.1% |
| | Crowdrun | -2.2% | -1.2% | -0.5% | -0.8% |
| | Hotel | -1.8% | -1.4% | -1.0% | -1.2% |
| | Pku_girl | -2.4% | -1.7% | -1.1% | -1.4% |
| | Reed | -2.1% | -1.5% | -1.3% | -1.6% |
| | 4k_seq1 | -3.7% | -2.7% | -2.0% | -2.2% |
| | Butterfly | -3.2% | -2.2% | -1.1% | -2.1% |
| 8K (7680×4320) | 8K_seq7 | -2.0% | -0.8% | -0.3% | -0.7% |
| | 8K_seq12 | -3.0% | -1.1% | -0.6% | -1.1% |
| | 8K_seq14 | -3.8% | -2.9% | -1.4% | -2.3% |
| | 8K_seq17 | -2.6% | -1.2% | -0.6% | -0.7% |
| | 8K_seq18 | -1.8% | -1.3% | -1.0% | -1.2% |
| | DS_store | -0.9% | -0.7% | -0.7% | -0.7% |
| | | | | | |
| | 2K average | -1.6% | -0.8% | -0.4% | -0.6% |
| | 4K average | -2.6% | -1.8% | -1.2% | -1.6% |
| | 8K average | -2.4% | -1.3% | -0.7% | -1.1% |
| | All average | -2.3% | -1.4% | -0.9% | -1.2% |

## IV. EXPERIMENTAL RESULTS

### A. Bitrate Reduction in the HEVC

The proposed intra prediction method is implemented in the HEVC reference software, HM-14.0 [22], on all Y, Cb and Cr components. The coding parameter configuration follows the HEVC common test conditions [23]. The main tier (QP = 22, 27, 32 and 37) is tested. Since we are more interested on high resolution video sequences, 2K to 8K sequences are tested. In this test, the kernel in Eq. (27) and the iteration numbers in Table I are used.

Since better intra-coded frames can also improve the coding efficiency of inter coding. Besides all intra (AI) coding, we also test our proposed method in random access (RA), low-delay B (LDB) and low-delay P (LDP) coding configures. Note that in these inter coding tests, all frames are coded. The BD-Bitrate [24] results for Y components are shown in Table II. The negative number means the bitrate is reduced compared with HM-14.0.

The gain of this proposed method is 2.3% on average and up to 3.8% in all intra (AI) coding, which is significant in the current HEVC coding standard. It is also clear that our

TABLE III
BEST ITERATION NUMBERS UNDER DIFFERENT MODES AND BLOCK SIZES

| Modes | 4×4 | 8×8 | 16×16 | 32×32 |
|---|---|---|---|---|
| 0 and 1 (Planar and DC) | 32 | 32 | 32 | 32 |
| 2 ~ 34 (Angular Modes) | 2 | 3 | 5 | 20 |

TABLE IV
BD-BITRATE REDUCTION COMPARED WITH OTHER METHODS IN AI CONFIGURATION ON SHORT SEQUENCESS

| | | [2] | [9] | **Ours with $K_1$** | **Ours with $K_2$** |
|---|---|---|---|---|---|
| Lower Resolution (416×240, 832×480) | BasketbalDrill | -0.3% | -0.2% | **-0.5%** | **-0.5%** |
| | BasketballPass | -0.4% | 0.0% | **-0.2%** | **-0.4%** |
| | BlowingBubbles | -0.4% | -0.2% | **-0.4%** | **-0.6%** |
| | PartyScene | -0.4% | -0.2% | **-0.4%** | **-0.5%** |
| | RaceHorse | -0.7% | -0.4% | **-1.0%** | **-1.0%** |
| | RaceHorseC | -0.8% | -0.6% | **-1.1%** | **-1.3%** |
| 2K (1920×1080) | Kimono | -1.2% | -0.9% | **-2.2%** | **-2.6%** |
| | ParkScene | -1.0% | -0.9% | **-1.5%** | **-1.7%** |
| | Cactus | -0.9% | -0.8% | **-1.3%** | **-1.6%** |
| | BasketballDrive | -2.2% | -1.0% | **-1.7%** | **-1.8%** |
| | BQTerrace | -0.5% | -0.3% | **-0.5%** | **-0.6%** |
| 4K (3840×2160, except hotel 4096×2048) | Chuno_s4 | -0.8% | -1.2% | **-2.0%** | **-2.1%** |
| | Chuno_s31 | -1.8% | -0.8% | **-3.1%** | **-3.3%** |
| | Crowdrun | -1.3% | -1.1% | **-2.0%** | **-2.3%** |
| | Hotel | -1.5% | -1.2% | **-1.8%** | **-1.8%** |
| | Pku_girl | -1.7% | -1.5% | **-2.2%** | **-2.4%** |
| | Reed | -1.6% | -1.7% | **-1.8%** | **-1.9%** |
| | 4k_seq1 | -2.8% | -2.2% | **-3.3%** | **-3.6%** |
| | Butterfly | -1.5% | -2.1% | **-2.6%** | **-2.8%** |
| 8K (7680×4320) | 8K_seq7 | -1.0% | -1.3% | **-2.0%** | **-1.9%** |
| | 8K_seq12 | -1.7% | -1.6% | **-2.7%** | **-2.9%** |
| | 8K_seq14 | -2.2% | -1.5% | **-4.2%** | **-4.0%** |
| | 8K_seq17 | -1.7% | -1.5% | **-2.3%** | **-2.5%** |
| | 8K_seq18 | -0.6% | -1.2% | **-1.8%** | **-1.8%** |
| | DS_store | -1.1% | -0.4% | **-0.7%** | **-0.8%** |
| | | | | | |
| | Encoding Time | 225% | 390% | **395%** | **446%** |
| | Decoding Time | 105% | 122% | **143%** | **186%** |
| | | | | | |
| | LR average | -0.5% | -0.3% | **-0.6%** | **-0.7%** |
| | 2K average | -1.2% | -0.8% | **-1.6%** | **-1.6%** |
| | 4K average | -1.5% | -1.5% | **-2.4%** | **-2.4%** |
| | 8K average | -1.4% | -1.3% | **-2.3%** | **-2.3%** |
| | All average | -1.2% | -1.0% | **-1.7%** | **-1.9%** |

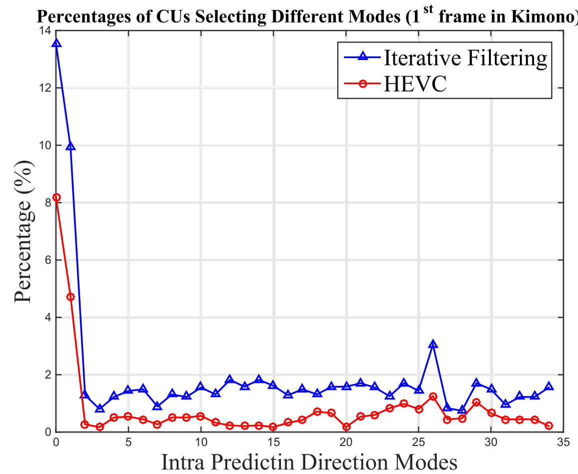
proposed method can also improve the coding gain in inter coding settings.

### B. Comparison With Other Methods

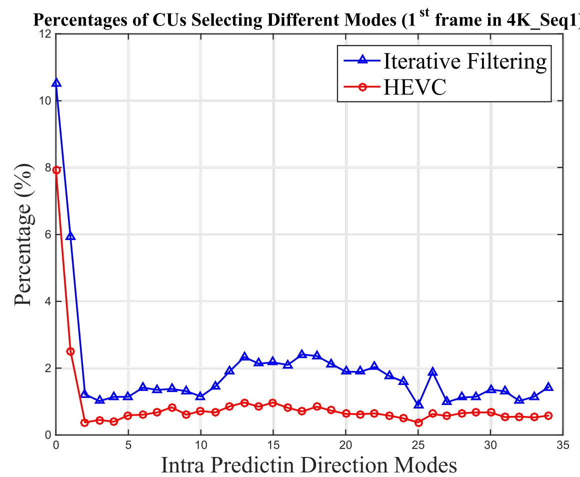
We compare following four methods as follows:

1) Recursive extrapolation intra prediction in [2];
2) PDE-based inpainting intra prediction in [9];
3) Our proposed method with following filtering kernel and the associated iteration numbers in Table III.

$$K_1 = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix};$$




(a) 1st frame of Kimono




(b) 1st frame of 4K_Seq1

Fig. 10. Percentage of CUs using iterative filtering and HEVC mode under different intra prediction directions in (a) Kimono and (b) 4K_Seq1.

4) Our proposed method with filtering kernel $K_2$ as in Eq. (27) and iteration numbers in Table I.

Note that, different from the testing in Section IV-A, we test more sequences with lower resolution (LR). Since our proposed method is a intra coding tool and coding performance on first few frames are very good approximation of the overall performance, we only test short sequences with first 10 frames for LR and 2K sequences and first 3 frames for 4K and 8K. All results are tested with AI configuration. The BD-Bitrate results are listed in Table IV.

Compared with other methods in [2] and [9], the proposed method outperforms those two on almost all testing sequences and the average improvement is about 0.5% to 0.7%. Comparing the results from kernel $K_1$ and kernel $K_2$, it can be seen that the gain is not sensitive to the selection of kernel, as long the proper iteration numbers are selected.

### C. Hit Ratio

To demonstrate the effectiveness of the proposed iterative filtering method, the hit ratio of this new intra prediction mode is presented in Fig. 10 and Fig. 11.

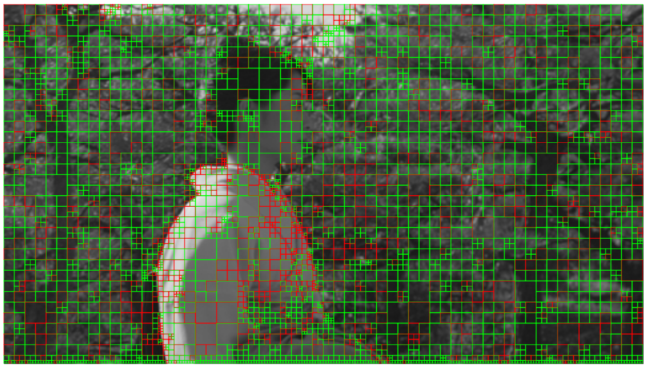


Fig. 11. Hit ratio in the first frame of Kimono. The green blocks indicate CUs using iterative filtering mode. The red blocks indicate CUs using HEVC intra prediction mode. (Best view in color).

TABLE V

BD-BITRATE REDUCTION ON SEPARATE MODES IN AI CONFIGURATION ON SHORT SEQUENCES

| | | Only DC | Only Planar | Only Angular |
|---|---|---|---|---|
| Lower Resolution (416×240, 832×480) | BasketbalDrill | 0.0% | 0.1% | -0.6% |
| | BasketballPass | 0.0% | 0.1% | -0.4% |
| | BlowingBubbles | 0.0% | 0.0% | -0.6% |
| | PartyScene | -0.1% | -0.1% | -0.4% |
| | RaceHorse | -0.3% | -0.2% | -1.1% |
| | RaceHorseC | -0.4% | -0.4% | -1.2% |
| 2K (1920×1080) | Kimono | -0.8% | -0.9% | -2.4% |
| | ParkScene | -1.0% | -0.9% | -1.6% |
| | Cactus | -0.4% | -0.5% | -1.5% |
| | BasketballDrive | -0.3% | -0.3% | -1.8% |
| | BQTerrace | -0.2% | -0.2% | -0.5% |
| 4K (3840×2160, except hotel 4096×2048) | Chuno_s4 | -0.6% | -0.6% | -2.0% |
| | Chuno_s31 | -0.9% | -1.4% | -3.0% |
| | Crowdrun | -0.8% | -0.8% | -2.1% |
| | Hotel | -0.5% | -0.4% | -1.8% |
| | Pku_girl | -0.8% | -0.7% | -2.3% |
| | Reed | -0.9% | -0.8% | -1.8% |
| | 4k_seq1 | -0.8% | -0.9% | -3.4% |
| | Butterfly | -0.3% | -0.3% | -2.8% |
| 8K (7680×4320) | 8K_seq7 | -0.3% | -0.4% | -1.9% |
| | 8K_seq12 | -0.7% | -0.6% | -2.8% |
| | 8K_seq14 | -0.7% | -1.4% | -3.8% |
| | 8K_seq17 | -0.8% | -0.6% | -2.4% |
| | 8K_seq18 | -0.5% | -0.6% | -1.8% |
| | DS_store | -0.2% | -0.2% | -0.8% |
| | | | | |
| | Encoding Time | 237% | 180% | 422% |
| | Decoding Time | 190% | 156% | 185% |
| | | | | |
| | LR average | -0.1% | -0.1% | -0.7% |
| | 2K average | -0.5% | -0.6% | -1.5% |
| | 4K average | -0.7% | -0.7% | -2.4% |
| | 8K average | -0.5% | -0.6% | -2.2% |
| | All average | -0.5% | -0.5% | -1.8% |

### *D. Separate Gain From DC, Planar and Angular Modes*

To investigate the separate gain from different modes, we test more cases with the proposed method applied on 1) only DC mode, 2) only Planar mode, and 3) only Angular modes. The filtering kernel follows Eq. (27) and the iteration number is in Table I. The results are shown in Table V.

TABLE VI

ITERATION NUMBERS FOR TESTS WITH DIFFERENT ITERATION NUMBERS

| | DC and Planar | Angular (4×4, 8×8, 16×6, 32×32) |
|---|---|---|
| Test 1 | 2 | 1, 1, 1, 1 |
| Test 2 | 4 | 2, 2, 2, 2 |
| Test 3 | 50 | 10, 20, 20, 30 |

TABLE VII

BD-BITRATE REDUCTION COMPARED WITH DIFFERENT ITERATION NUMBERS IN AI CONFIGURATION ON SHORT SEQUENCESS

| | | Test 1 | Test 2 | Test 3 |
|---|---|---|---|---|
| Lower Resolution (416×240, 832×480) | BasketbalDrill | -0.3% | -0.4% | -0.4% |
| | BasketballPass | -0.3% | -0.3% | -0.1% |
| | BlowingBubbles | -0.5% | -0.5% | -0.3% |
| | PartyScene | -0.5% | -0.5% | -0.2% |
| | RaceHorse | -0.9% | -1.0% | -0.7% |
| | RaceHorseC | -0.9% | -1.2% | -0.9% |
| 2K (1920×1080) | Kimono | -0.5% | -0.7% | -2.8% |
| | ParkScene | -1.0% | -1.4% | -1.4% |
| | Cactus | -1.1% | -1.2% | -1.1% |
| | BasketballDrive | -1.4% | -1.5% | -1.5% |
| | BQTerrace | -0.9% | -0.7% | -0.3% |
| 4K (3840×2160, except hotel 4096×2048) | Chuno_s4 | -1.7% | -2.0% | -1.4% |
| | Chuno_s31 | -0.9% | -1.2% | -3.3% |
| | Crowdrun | -1.4% | -1.8% | -1.8% |
| | Hotel | -1.6% | -1.7% | -1.5% |
| | Pku_girl | -1.2% | -1.6% | -1.9% |
| | Reed | -0.7% | -1.1% | -1.8% |
| | 4k_seq1 | -1.4% | -2.1% | -3.2% |
| | Butterfly | -1.6% | -2.2% | -1.9% |
| 8K (7680×4320) | 8K_seq7 | -1.4% | -1.7% | -1.4% |
| | 8K_seq12 | -1.8% | -2.4% | -2.1% |
| | 8K_seq14 | -2.2% | -3.0% | -3.6% |
| | 8K_seq17 | -2.0% | -2.4% | -1.8% |
| | 8K_seq18 | -1.0% | -1.2% | -1.5% |
| | DS_store | -1.0% | -1.0% | -0.5% |
| | | | | |
| | Encoding Time | 239% | 262% | 679% |
| | Decoding Time | 117% | 124% | 256% |
| | | | | |
| | LR average | -0.6% | -0.6% | -0.4% |
| | 2K average | -1.0% | -1.1% | -1.4% |
| | 4K average | -1.3% | -1.6% | -2.1% |
| | 8K average | -1.6% | -1.9% | -1.8% |
| | All average | -1.1% | -1.3% | -1.4% |

The result shows that applying only on DC and only on Planar achieve the similar gain, both are about 0.5% on average. Applying on only angular modes can achieve a gain of 1.8%, which is close to the gain on all modes. One reason that angular modes contribute most of the gain is that most blocks use angular modes. Another reason is that the smoothed angular blocks sometimes are similar to smoothed DC or smoothed Planar predicted blocks, so that some blocks originally using the DC or Planar modes would change to the angular modes. This is also the reason why the gain on separate modes are not additive.

### *E. Effects of Iteration Numbers*

The number of convolution iterations affects the computation complexity. To investigate the coding performance of the

TABLE VIII
TRAINED PARAMETERS IN FIRST-ORDER MARKOV 2D DIRECTIONAL MODEL

| Mode | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| $\alpha$ | 0° | 0° | 45° | 39.1° | 33.3° | 28.0° | 22.1° |
| $\rho$ | 0.9927 | 0.9942 | 0.9909 | 0.9945 | 0.9985 | 0.9961 | 0.9930 |
| $\eta$ | 1.0406 | 1.0167 | 2.0448 | 3.4841 | 16.9953 | 8.6324 | 6.0968 |
| Mode | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| $\alpha$ | 15.7° | 8.8° | 3.6° | 0° | -3.6° | -8.8° | -15.7° |
| $\rho$ | 0.9904 | 0.9991 | 0.9951 | 0.9911 | 0.9915 | 0.9919 | 0.9924 |
| $\eta$ | 4.4657 | 38.9562 | 21.5611 | 4.1660 | 4.2770 | 4.3879 | 4.4989 |
| Mode | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| $\alpha$ | -22.1° | -28.0° | -33.3° | -39.1° | -45° | 129.1° | 123.3° |
| $\rho$ | 0.9928 | 0.9932 | 0.9936 | 0.9940 | 0.9905 | 0.9934 | 0.9942 |
| $\eta$ | 4.6099 | 4.7209 | 4.8318 | 4.9428 | 2.6771 | 4.1971 | 5.0094 |
| Mode | 21 | 22 | 23 | 24 | 25 | 26 | 27 |
| $\alpha$ | 118.0° | 112.1° | 105.7° | 98.9° | 93.6° | 90° | 86.4° |
| $\rho$ | 0.9951 | 0.9953 | 0.9959 | 0.9954 | 0.9919 | 0.9944 | 0.9941 |
| $\eta$ | 5.8217 | 6.3431 | 8.5213 | 7.1970 | 3.8479 | 3.5534 | 6.8005 |
| Mode | 28 | 29 | 30 | 31 | 32 | 33 | 34 |
| $\alpha$ | 81.1° | 74.3° | 67.9° | 62.0° | 56.7° | 50.9° | 45° |
| $\rho$ | 0.9950 | 0.9958 | 0.9967 | 0.9973 | 0.9983 | 0.9928 | 0.9910 |
| $\eta$ | 6.5934 | 7.4613 | 8.9982 | 11.1011 | 14.4937 | 2.8417 | 2.3281 |

proposed method with different iterations, more tests on the kernel as in Eq. (27) are conducted with iteration numbers in Table VI.

The BD-Bitrate results are shown in Table VII.

The result shows that if the iteration number is too small or too larger, the performance is degraded (compared to the best result of 1.9%, in Table IV, last column). For the complexity, it is clear that a smaller iteration number has less complexity.

### F. Encoding and Decoding Complexity

According to the encoding and decoding time in Table IV, the proposed new intra prediction mode increases the encoder time by about 400% of the anchor and the decoding time by about 150% of the anchor, in the All-Intra configuration. The additional complexity mainly comes from the iterative filtering and the extra mode decision. By reducing the iteration number, the additional encoding and decoding time can be reduced. In Table IV Test 2, the complexity is reduce to 262% (encoding) and 124% (decoding). Note that this new intra prediction mode can be done in parallel to current HEVC mode. More work will be done to reduce the complexity. Also, in most practical applications, both inter coding and intra coding are used. Since inter coding requires much more computations, the overall impact on the complexity is not as high.

## V. CONCLUSION

This paper has two contributions: 1) it discusses the intra prediction weights in the situation when the predicted samples have low correlation with the reference sample and the reference samples have deviation, and 2) proposes a novel intra prediction scheme using iterative filtering, which is suitable for cases in 1). The proposed method is used as an extra intra prediction mode in addition to the copying-based HEVC intra prediction scheme. The theoretical coding gain is used to compare the proposed scheme and the current HEVC intra prediction scheme. It shows that better gain is achieved by the iterative filtering method when a proper iteration number is selected. The experimental results in the HEVC reference software also confirms the effectiveness. The BD-bitrate reduction is up to 3.8% and 2.3% on average. One of our future work is to find a better way to design the kernels, e.g., adaptive to the block size, to have better results and less computation complexity.

## APPENDIX A

The 2D directional Gaussian Markov model in Eq. (16) can be rewritten in a linear format,

$$s = \beta_1 t_1 + \beta_2 t_2, \tag{28}$$

where

$$\begin{aligned} s &= \log^2 E(x_{i,j} x_{p,q}) \\ \beta_1 &= \log^2 \rho \\ t_1 &= d_1^2(\alpha) \\ \beta_2 &= \eta^2 \bullet \log^2 \rho \\ t_2 &= d_2^2(\alpha) \end{aligned}$$

Given the auto-correlation matrix of the mode and the dominating direction angle $\alpha$, the values s, $t_1$ and $t_2$ are known and the $\beta_1$ and $\beta_2$ can be derived from a linear regression, then the $\rho$ and $\eta$ are calculated. The training process has following steps:

1. Select the first frame from six sequences: Kimono, Cactus, BasketballDrive, BQTerrace, 4K_Seq1 and Butterfly.
2. For each image, it is divided into non-overlapped $8 \times 8$ small patches and the intra mode of each patch is determined via minimizing the SAD between the prediction and the image patch.
3. All patches with the same mode are classified as the "observations" of that mode and the auto-correlation matrix is calculated.
4. Use the least square method to find the $\beta_1$ and $\beta_2$, and calculate $\rho$ and $\eta$.

For some of these modes, when the trained parameter is not a real number, e.g., when $\beta_1$ is negative, parameters are linearly interpolated with parameters of its neighboring mode. The final training result is shown in Table V. For simplicity, in analysis within this paper, a typical set of parameters $\rho = 0.99$ and $\eta = 5$ (1 for DC and Planar modes) are used.

## APPENDIX B

For the distorted reference, the optimal weights are derived by Eq. (26). Given $N$ reference samples, the following approximation holds when $\sigma^2$ is large and $\rho -> 1$.

$$E(\bar{A}\bar{A}^T) \approx \begin{pmatrix} 1+\sigma^2 & 1 & \cdots & 1 \\ 1 & 1+\sigma^2 & \cdots & \vdots \\ \vdots & \vdots & \ddots & 1 \\ 1 & \cdots & 1 & 1+\sigma^2 \end{pmatrix}_{N \times N}$$

$$E(b_i \bar{A}) \approx \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}_{N \times 1}$$

Hence the optimal weights are approximated as

$$w_{i,opt} \approx \begin{pmatrix} 1+\sigma^2 & 1 & \cdots & 1 \\ 1 & 1+\sigma^2 & \cdots & \vdots \\ \vdots & \vdots & \ddots & 1 \\ 1 & \cdots & 1 & 1+\sigma^2 \end{pmatrix}^{-1} \bullet \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$$
$$= \frac{1}{\sigma^2 + N} \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$$

So when the error is large, the optimal weights of each reference pixels are the same.

## REFERENCES


[1] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[2] Y. Chen, J. Han, and K. Rose, "A recursive extrapolation approach to intra prediction in video coding," in *Proc. ICASSP*, 2013, pp. 1734–1738.

[3] S. Li, Y. Chen, J. Han, T. Nanjundaswamy, and K. Rose, "Rate-distortion optimization and adaptation of intra prediction filter parameters," in *Proc. ICIP*, 2014, pp. 3146–3150.

[4] F. Kamisli, "Recursive prediction for joint spatial and temporal prediction in video coding," *IEEE Signal Process. Lett.*, vol. 21, no. 6, pp. 732–736, Jun. 2014.

[5] N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*. Cambridge, MA, USA: MIT Press 1949.

[6] J. Xu, F. Wu, and W. Zhang, "On the coding gain of intra-predictive transforms," in *Proc. ICIP*, 2009, pp. 13–16.

[7] L. Zhang, X. Zhao, S. Ma, Q. Wang, and W. Gao, "Novel intra prediction via position-dependent filtering," *J. Vis. Commun. Image Represent.*, vol. 22, no. 8, pp. 687–696, 2011.

[8] X. Qi, T. Zhang, F. Ye, A. Men, and B. Yang, "Intra prediction with enhanced inpainting method and vector predictor for HEVC," in *Proc. ICASSP*, 2012, pp. 1217–1220.

[9] Y. Zhang and Y. Lin, "Improving HEVC intra prediction with PDE-based inpainting," in *Proc. APSIPA*, 2014, pp. 1–5.

[10] J. Han, A. Saxena, V. Melkote, and K. Rose, "Jointly optimized spatial prediction and block transform for video and image coding," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1874–1884, Apr. 2012.

[11] X. Chen, J.-N. Hwang, C.-N. Lee, and S.-I. Chen, "A near optimal QoE-driven power allocation scheme for scalable video transmissions over MIMO systems," *IEEE J. Sel. Top. Signal Process.*, vol. 9, no. 1, pp. 76–88, Feb. 2015.

[12] M. Budagavi and D.-K. Kwon, *AHG8: Video Coding Using Intra Motion Compensation*, document JCTVC-M0350, Joint Collaborative Team on Video Coding (JCT-VC), 2013.

[13] H. Chen, Y.-S. Chen, M.-T. Sun, A. Saxena, and M. Budagavi, "Improvements on intra block copy in natural content video coding," in *Proc. ISCAS*, 2015, pp. 2772–2775.

[14] T. Zhang, H. Chen, M.-T. Sun, D. Zhao, and W. Gao, "Hybrid angular intra/template matching prediction for HEVC intra coding," in *Proc. Vis. Commun. Image Process. (VCIP)*, 2015, pp. 1–4.

[15] C. Lan, X. Peng, J. Xu, and F. Wu, *Intra and Inter Coding Tools for Screen Contents*, document JCTVC-E145, Joint Collaborative Team on Video Coding (JCT-VC), 2011.

[16] H. Chen, A. Saxena, and F. Fernandes, "Nearest-neighbor intra prediction for screen content video coding," in *Proc. ICIP*, 2014, pp. 3151–3155.

[17] H. Chen and B. Zeng, "New transforms tightly bounded by DCT and KLT," *IEEE Signal Process. Lett.*, vol. 19, no. 6, pp. 344–347, Jun. 2012.

[18] T. Zhang, H. Chen, M.-T. Sun, A. Saxena, D. Zhao, and W. Gao, "Adaptive transform with HEVC intra coding," in *Proc. Asia-Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA)*, 2015, pp. 388–391.

[19] H. Chen and B. Zeng, "Design of low-complexity, non-separable 2-D transforms based on butterfly structures," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2012, pp. 2921–2924.

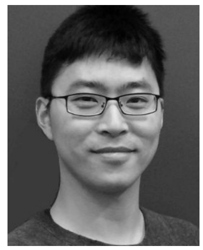
[20] N. S. Jayant and P. Noll, *Digital Coding of Waveforms: Principles and Applications to Speech and Video*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1984.

[21] T. Natarajan and N. Ahmed, "Performance evaluation for transform coding using a nonseparable covariance model," *IEEE Trans. Commun.*, vol. 26, no. 2, pp. 310–312, Feb. 1978.

[22] *Hevc Test Model (hm-14.0)*, accessed on Apr. 11, 2014. [Online]. Available: http://hevc.kw.bbc.co.uk/git/w/jctvc-hm.git/commit/1204fc83b1f917237624e9ec8c3c4d7bebab58d5

[23] F. Bossen, *Common HM Test Conditions and Software Reference Configurations (JCTVC-L1100)*, document JCT-VC, 2013.

[24] G. Bjontegaard, *Calculation of Average PSNR Differences Between RD Curves*, document VCEG-M33 ITU-T Q6/16, 2001.

**Haoming Chen** received the B.S. degree in electrical engineering from Nanjing University, China, in 2010, and the M.Phil. degree in electrical engineering from the Hong Kong University of Science and Technology in 2012. He is currently pursuing the Ph.D. degree with the Electrical Engineering Department, University of Washington. His research interests include video coding, multimedia communication, and image processing. He has interned with Samsung Research America, Richardson, TX, in 2013 summer, and Apple, Cupertino, CA, in 2015 summer. He was a recipient of the top 10% paper at the IEEE ICIP 2014 and the Student Best Paper Award Finalist at the IEEE ISCAS 2012.

**Tao Zhang** received the B.S. degree in computer science from Xidian University, Xi'an, China, in 2010, and the M.S. degree in computer science from the Harbin Institute of Technology (HIT), Harbin, China, in 2012.

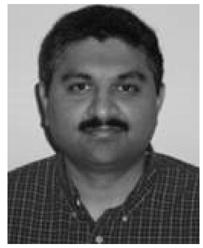
He is currently pursuing the Ph.D. degree in computer science with HIT. From 2012 to 2013, he was with Microsoft Research Asia, Beijing, China, as an Intern. From 2014 to 2015, he was a Visiting Student with Information Processing Laboratory, University of Washington, Seattle, WA, USA. His current research interests include image and video coding.

**Ming-Ting Sun** (S'79–M'81–SM'89–F'96) received the B.S. degree in electrical engineering from National Taiwan University in 1976, and the Ph.D. degree in electrical engineering from the University of California at Los Angeles, in 1985.

He joined the University of Washington in 1996, where he is a Professor. He was the Director of Video Signal Processing Research with Bellcore. He has been a Chair/Visiting Professor with Tsinghua University, Tokyo University, National Taiwan University, National Cheng Kung University, National Chung Cheng University, National Sun Yat-sen University, the Hong Kong University of Science and Technology, and National Tsing Hua University. He co-edited a book entitled *Compressed Video Over Networks*. He has guest-edited 12 special issues for various journals and given keynotes for several international conferences. His research interests include video and multimedia signal processing, and transport of video over networks.

He holds 13 patents and has published over 200 technical papers, including 17 book chapters in the area of video and multimedia technologies. He has served on many prestigious awards committees. He was a Technical Program Co-Chair for several conferences, including the International Conference on Multimedia and Expo (ICME) 2010. He received the IEEE CASS Golden Jubilee Medal in 2000, and was the General Co-Chair of the Visual Communications and Image Processing 2000 Conference. He was the Editor-in-Chief of the IEEE TRANSACTIONS ON MULTIMEDIA and a Distinguished Lecturer of the Circuits and Systems (CAS) Society from 2000 to 2001. He was the Editor-in-Chief of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (TCSVT) from 1995 to 1997. He served as the Chair of the Visual Signal Processing and Communications Technical Committee of the IEEE CAS Society from 1993 to 1994. He received the TCSVT Best Paper Award in 1993. From 1988 to 1991, he was the Chairman of the IEEE CAS Standards Committee and established the IEEE Inverse Discrete Cosine Transform Standard. He received an Award of Excellence from Bellcore for his work on the digital subscriber line in 1987. He will be a General Co-Chair of the IEEE ICME 2016, Seattle.

**Ankur Saxena** (S'06–M'09) was born in Kanpur, India, in 1981. He received the B.Tech. degree in electrical engineering from IIT Delhi, India, in 2003, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of California at Santa Barbara, CA, USA, in 2004 and 2008, respectively. He interned with the Fraunhofer Institute of X-Ray Technology, Erlangen, Germany, and the NTT DoCoMo Research Laboratory, Palo Alto, CA, USA, in the summers of 2002 and 2007, respectively. From 2010 to 2016, he was with Samsung Research America-Dallas, TX, USA, where he was Samsung's official ISO/IEC delegate for MPEG and JCT-VC and contributed to the standardization of HEVC and CDVS standardizations. He is currently with Nvidia, Santa Clara, CA. His current research interests include source coding, image and video processing, virtual reality, and picture quality enhancement. He was a recipient of the President Work Study Award during his Ph.D. studies, the Best Student Paper Finalist at ICASSP in 2009, the Samsung Best Paper Award in 2011, and the IEEE Signal Processing Society Young Author Best Paper Award in 2015. He has been a Co-Chair for seven core experiments in HEVC/CDVS, and has over 40 publications in peer-reviewed journals and conferences, and over 45 standard contributions.



**Madhukar Budagavi** (M'98–SM'06) received the B.E. degree in electronics engineering from the National Institute of Technology, Trichy, India, in 1991, the M.S. degree in electrical engineering from the Indian Institute of Science, Bangalore, India, in 1994, and the Ph.D. degree in electrical engineering from Texas A&M University, College Station, TX, USA, in 1998.

He was with the Embedded Processing Research and Development Center, Texas Instruments, Dallas, TX, USA, from 1998 to 2014. He was a Co-Editor of the book entitled *High Efficiency Video Coding (HEVC): Algorithms and Architectures* (Springer, 2014). He has been an active participant in the standardization of the HEVC (ITU-T H.265 | ISO/IEC 23008-2) next-generation video coding standard by the JCT-VC committee of ITU-T and ISO/IEC. Within the JCT-VC committee, he has chaired and co-chaired technical subgroup activities on coding tools for HEVC, scalable HEVC, and the screen content coding HEVC. He has authored several book chapters, and journal and conference papers. He is currently a Senior Director and the Multimedia Standards Team Leader with the Standards and Mobility Innovation Laboratory, Samsung Research America, Dallas, and represents Samsung in multimedia standards activities in ITU-T, ISO/IEC, UHD Alliance, and the Society of Motion Picture and Television Engineers. His experience includes research and development of compression algorithms, image and video processing, video codec system-on-chip architecture, embedded vision, 3D graphics, speech coding, and embedded software implementation and prototyping.