Short Distance Intra Prediction of Screen Content in Versatile Video Coding (VVC)

Mohsen Abdoli, Felix Henry, Patrice Brault, Pierre Duhamel, Frédéric Dufaux

HAL Id: hal-01876742
https://hal.science/hal-01876742

Short Distance Intra Prediction of Screen Content in Versatile Video Coding (VVC)

Mohsen Abdoli, Félix Henry, Patrice Brault, *Senior Member*, Pierre Duhamel, *Fellow, IEEE*
and Frédéric Dufaux, *Fellow, IEEE*

Abstract—A novel intra prediction algorithm is proposed to improve the coding performance of screen content for the emerging Versatile Video Coding (VVC) standard. The algorithm, called In-Loop Residual coding with Scalar Quantization (ILR-SQ), employs in-block pixels as reference rather than the regular out-block ones. To this end, an additional in-loop residual signal is used to partially reconstruct the block at the pixel level, during the prediction. The proposed algorithm is essentially designed to target high detail textures, where deep block partitioning structure is required. Therefore, it is implemented to operate on 4×4 blocks only, where further block split is not allowed and the standard algorithm is still unable to properly predict the texture. Experiments in the Joint Exploration Model (JEM) reference software show that the proposed algorithm brings a BD-rate gain of 13% on synthetic content, with a negligible computational complexity overhead at both encoder and decoder sides.

Index Terms—Intra prediction, VVC, Screen content coding

I. INTRODUCTION

Applications of synthetic videos, such as remote screen sharing, online gaming, animated movies, virtual reality, have become more popular in the past few years. This has drawn the attention of standardization committees to dedicate specific compression tools to this type of content. In the High Efficiency Video Coding standard, i.e. H.265/HEVC, these tools have formed an extension called HEVC Screen Content Coding (SCC) [1]. This extension has brought significant coding gain on synthetic videos by using four major tools: Intra Block Copy (IBC) [2], Palette mode [3], Adaptive Color Transform (ACT) [4] and Adaptive Motion Vector Resolution (AMVR) [5].

In the development of the next generation video codec, the Versatile Video Coding (VVC), a new standardization action has started in late 2015. For this purpose, the ITU-T Video Coding Expert Group (VCEG) and the ISO Motion Picture Expert Group (MPEG) have jointly initiated a committee, called Joint Video Exploration Team (JVET). The agenda of this committee is to significantly exceed the performance of HEVC and explore video compression capabilities beyond it. The outcome of this action, expected to be standardized in 2020, is supposedly a universal replacement for HEVC and all of its extensions, including SCC [6].

M. Abdoli and F. Henry are with the Orange Labs, Cesson-Sévigné, France. E-mails: {mohsen.abdoli and felix.henry}@orange.com

P. Brault, P. Duhamel and F. Dufaux are with L2S, CNRS - CentraleSupelec - Université Paris-Sud, Gif-sur-Yvette, France. E-mails: {patrice.brault, pierre.duhamel, frederic.dufaux}@l2s.centralesupelec.fr.

A. New tools and features

JVET consists of an exploration phase and a standardization phase. In each phase, different reference softwares are used to evaluate the performance of new tools. During the exploration phase, the Joint Exploration Model (JEM) was developed, in order to reach maximum performance before the actual standardization phase. In early 2018, two other reference software, the VVC Test Model (VTM) and the BenchMark Set (BMS), were released for the standardization phase. The VTM is mostly based on HEVC test Model (HM), with many tools removed and a few added, and improves HEVC by 12% in compression performance. The BMS and JEM, however, benefit from quite similar tools and both bring about 30% to 35% compression gain over the HM with different complexity overhead [7]. Here is the list of major tools and contributions that are currently under consideration for the VVC:

- Partitioning: Nested multi-type tree including quad-tree, binary-tree and ternary-tree [8] [9] [10] [11].
- Inter prediction: Bi-directional Optical Flow (BIO [12]), Overlapped Block Motion Compensation (OBMC) [8].
- Intra prediction: 67 Intra Prediction Modes (IPM), Position Dependent Prediction Combination (PDPC [13]), Cross-component linear model prediction [8].
- Transform coding: Large block-size transforms with high-frequency zeroing, Non-Separable Secondary Transforms (NSST [14]), Adaptive Multiple Transforms (AMT [15]), Adaptive Multiple Core Transforms.
- Filters: Adaptive Loop Filter (ALF), Four-tap intra prediction filter, Bilateral in-loop filtering [8].

B. Distant reference inaccuracy of intra prediction

To model angular textures more accurately, the size of the IPM set is doubled in the JEM. This new set shares the DC and Planar IPMs with HEVC, but extends the set of angular modes from 33 to 65 IPMs, thus covering the same angular range with double precision. Although the extended IPM set has brought a significant coding gain, it is still unable to properly predict high detail content with complex textures [16]. This is due to the low correlation between distant out-block references and in-block prediction pixels. This problem, called the Distant Reference Inaccuracy (DRI) in the rest of this letter, may be eased by further block partitioning. However, deeper block partitioning always comes with the cost of extra block level syntax elements to transmit. More importantly, the DRI often happens at the 4×4 blocks, where further partitioning is not allowed. Fig. 1 depicts some texture
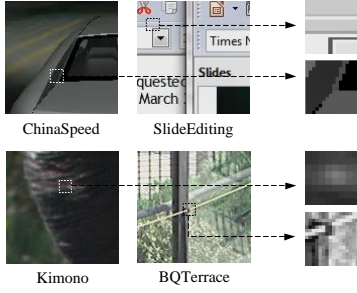
Fig. 1. Example in-block content changes from synthetic (ChinaSpeed and SlideEditing) and natural (Kimono and BQTerrace) sequences.

examples from natural and synthetic video sequences. As can be seen, in both cases, the local content change makes the in-block pixels quite uncorrelated to their out-block references. However, due to the solid edges in the synthetic videos, the in-block pixel correlation is stronger than natural content.

Previous research works have studied a similar problem. In Combined Intra Prediction (CIP), an additional in-block pixel prediction is combined with the traditional prediction using a weighted average [17, 18]. In Short Distance Intra Prediction (SDIP), use of further block splitting is proposed to achieve excessive thin block in order to reduce the distance from reference pixels [19]. Finally, Residual Differential Pulse-Code Modulation (RDPCM) uses residual prediction with immediate neighbors inside the same block, in order to further decorrelate signal [20]–[22]. From this aspect, the proposed algorithm shares a common concept with RDPCM. However, instead of residual signal, it performs the DPCM-based prediction on the pixel values directly.

In this letter, a novel intra prediction algorithm is proposed, which is well suited for synthetic blocks presenting the DRI problem. The algorithm, called In-Loop Residual coding with Scalar Quantization (ILR-SQ), is integrated as an additional tool in the JEM and its usage requires signaling by the encoder. The ILR-SQ uses in-block reference pixels to apply a DPCM-based prediction. To exploit in-block references, an additional residual is coded. From this perspective, the proposed idea shares the same objective as our previous research that adopts a Vector Quantization (VQ) technique, called ILR-VQ [16]. It is also important to note that the fact that ILR-SQ is designed and implemented within the framework of VVC does not harm its generalizability. In other words, ILR-SQ can be adopted by all hybrid block-based video coding standards.

The rest of this letter is organized as follows. In section II, the ILR-SQ algorithm and its different modules are described in detail. In section III, the block level algorithm competition between the JEM and the proposed ILR-SQ is explained. In section IV, experimental results of embedding the ILR-SQ in the JEM are presented. Finally, section V concludes the letter.

II. INTRA PREDICTION BY IN-LOOP RESIDUAL CODING AND SCALAR QUANTIZATION

In this section, the in-block prediction to address the DRI problem is introduced. For this purpose, first, the main obstacle

for using in-block reference pixels is discussed. Then, block prediction by ILR-SQ is described in detail. Finally, the proposed residual coding scheme is presented.

A. Data Dependency for In-Block References

The major obstacle for using in-block references is a data dependency between prediction and residual signals. More precisely, at the time of block prediction, the closest fully reconstructed references are the regular out-block pixels from previous blocks, as used in the conventional intra prediction. Therefore, exploiting in-block references requires reconstructing them beforehand or during the prediction. This makes the residual a prerequisite to the prediction phase, while the residual signal is traditionally accessible after the prediction phase is finished at the block level. In other words, the prediction signal depends on the residual signal and conversely, the residual signal also depends on the prediction signal. In order to address this problem, an additional In-Loop Residual (ILR) signal is computed during the prediction and transmitted to the decoder alongside the regular out-loop residual (OLR).

The ILR signal is responsible for pixel reconstruction right after prediction at the pixel level. This enables in-block references for the prediction of next pixels. In order to draw a distinction from the regular pixel reconstruction by the OLR, this step is called pixel "correction" in the rest of this letter.

B. In-Block Pixel Prediction and ILR Calculation

Without loss of generality, let us assume that we are given a square block of size N with its out-block reference pixels. We are also given a scan order (e.g. raster scan) to determine the order of pixel prediction inside the block. Let the matrix $O_{N \times N}$ represent the input original signal, $P_{N \times N}$ the output prediction signal and $I_{N \times N}$ the decodable ILR signal to be calculated during the prediction. Also, let $R_{(N+1) \times (N+1)}$ be the concatenation of the out-block reconstructed and the in-block corrected references. For simplicity, the first row and column of matrix R contain the out-block reference pixels and are indexed by -1 instead of 0. From these signals, the entire O matrix as well as the first row and first column of R matrix are filled prior to the prediction phase whereas the remaining matrices are calculated during the prediction phase.

Starting from pixel index ($x = 0, y = 0$), the goal is to fill P by predicting all pixels in the block, meanwhile calculating elements in matrix R. For this purpose, the algorithm traverses the pixels according to the scan pattern and performs the following five steps at each pixel index (x, y):

1) Access three reference pixels at left $A = R_{(x-1,y)}$, top $B = R_{(x,y-1)}$ and top-left $C = R_{(x-1,y-1)}$.
   According to the above descriptions, these references can either be out-block or in-block pixels.
2) Predict the pixel by the LOw COmplexity LOssless COmpression for Images (LOCO-I) function [23]:

$$P_{(x,y)} = \begin{cases} min(A, B), & \text{if } C \geq max(A, B). \\ max(A, B), & \text{if } C \leq min(A, B). \\ A + B - C, & \text{Otherwise.} \end{cases} \quad (1)$$
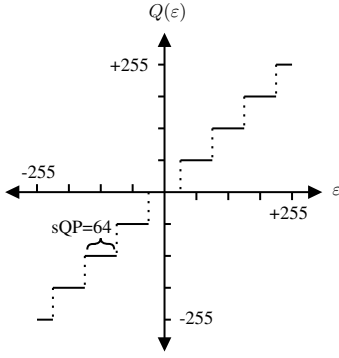
Fig. 2. Linear scalar quantization function with sQP= 64.

Fig. 3. An example 4×4 quantized ILR coefficient block at left and the visual representation of its truncated unary codes at right.

The LOCO-I function was first adopted in the JPEG-LS image compression standard. It aims at pixel-level edge detection in the vertical and horizontal directions (i.e. the first two conditions of Eq. 1, respectively). Moreover, if no edge is detected, it applies a linear filter on the references (i.e. the last condition of Eq. 1).

3) Calculate the error:

$$\varepsilon = O_{(x,y)} - P_{(x,y)}. \tag{2}$$

4) Calculate ILR by quantizating the error in the spatial domain:

$$I_{(x,y)} = Q(\varepsilon). \tag{3}$$

Here, Q is a linear scalar quantizer with the parameter sQP shown in Fig. 2, where sQP determines the compression rate of the ILR. This parameter is a function of the regular quantization parameter QP which is applied in the transform domain coefficients. For this purpose, the quantization scheme used in the Transform Skip Mode (TSM) of HEVC is adopted. In the TSM, due to a similar problem of residual quantization in the pixel domain, proper normalization and shift is performed to adapt the quantization step size in the pixel domain [24].
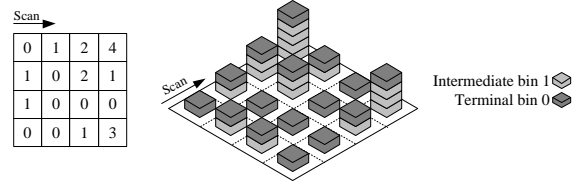
5) Correct the pixel by adding the ILR value to the predicted value:

$$R_{(x,y)} = P_{(x,y)} + I_{(x,y)}. \tag{4}$$

By performing the above five steps on the pixel at index (x,y), the corresponding values of all matrices P, I and R are determined. Therefore, from now on, the corrected pixel at $R_{(x,y)}$ can act as an in-block reference during remaining iterations of the prediction phase.

C. ILR Coding

To reproduce the same prediction signal at the decoder side, the ILR signal in I must be transmitted. For this purpose, individual ILR values in I are divided by sQP to produce integer symbols. These amplitude symbols are then binarized and compressed, while their signs are bypassed-coded.

1) Binarization: After thoroughly studying the distribution of numerous ILR signals in different situations, the truncated unary representation is selected as the binarization scheme of ILR coefficient magnitudes. To perform the binarization, the matrix I is scanned. For each coefficient with magnitude X, the truncated unary code is produced as a sequence of $X + 1$ bins, including X intermediate bins of '1', followed by a single bin of terminal '0'. Fig. 3 visualizes the binarization process for a 4×4 block example.

2) Compression: After the magnitude binarization is finished for the entire block, the bins are compressed using Context-Adaptive Binary Arithmetic Coding (CABAC), with suitable context models. For this purpose, the maximum magnitude is set to 15, limiting the number of dedicated CABAC contexts to 16. In other words, all bins after the first 16 bins of unary codes would share the last CABAC context for encoding and decoding.

III. BLOCK-LEVEL COMPETITION ALGORITHM

The ILR-SQ algorithm targets high detail blocks with the DRI problem that the state-of-the-art is unable to efficiently address. Such blocks usually occupy a noticeable portion of the picture in synthetic video content and the IRL-SQ is able to reduce their rate-distortion cost significantly. However, the other portion of the picture which contain smoother and more regular texture is still coded more efficiently by the conventional intra prediction algorithm than the ILR-SQ. Therefore, we integrate the ILR-SQ alongside the regular algorithm and allow the rate-distortion optimization to choose the best one as summarized in Fig. 4. Such flexibility of having multiple options at the block level imposes both rate and complexity overhead. Regarding the rate, the encoder has to transmit one flag per block to explicitly indicate the selected algorithm. For this purpose, this syntax element is compressed by CABAC and transmitted to the decoder before all the other block-level syntax elements. Moreover, deciding about the flag value for each block requires the encoder to actually apply both JEM and ILR-SQ algorithms for each eligible block. This increases the complexity of the rate-distortion optimization (RDO) loop, in which all possible combinations of the encoding tools are evaluated.

Fig. 5 visualizes result of the competition algorithm in two sequences. As expected, complex textures with high detail edges in each sequence, are coded by ILR-SQ, where plain areas and regular patterns are coded by the JEM algorithm.
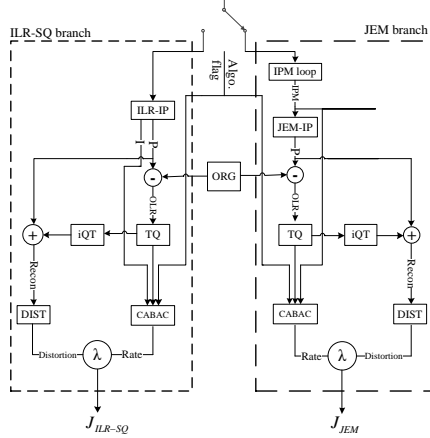
Fig. 4. The intra prediction competition algorithm at the block level. Here, *OLR* is the regular out-loop residual and *Recon* is the reconstructed signal.

Fig. 5. Texture coded by ILR-SQ (bright) and JEM (dark).

ChinaSpeed   SlideEditing

IV. RESULTS

In this section, experimental results are provided to evaluate performance of ILR-SQ. Due to the fact that until the last stages of this research, only the JEM was available, this reference software was used as the anchor reference for the implementation. However, since the JEM and the BMS references share most of their tools, the same level of performance is also expected in the BMS.

For the test, 14 synthetic video sequences from the SCC Common Test Conditions (CTC) [25] and Class F of the JVET CTC [26] were used. As recommended in both CTCs, all frames of the sequences are coded. Table I compares the average selection rate of ILR-SQ blocks at different QPs by the JEM5.0.1. As can be seen, in all QPs, a significant portion of the blocks are encoded by the ILP-SQ. This proves the claim that the conventional intra prediction of the JEM is not efficient for synthetic textures.

TABLE I
AVERAGE SELECTION RATE OF THE ILR-SQ AT DIFFERENT QPs.

	Quantization Parameter			
	22	27	32	37
Among 4×4 blocks	51%	73%	75%	77%
Among all blocks	29%	41%	39%	38%

Table II compares the coding performance of ILR-SQ against the JEM in all intra (AI) mode (4×4 blocks only). As another benchmark, ILR-SQ is also compared to its previous

TABLE II
PERFORMANCE OF THE PROPOSED ILR-SQ AND THE ILR-VQ AGAINST JEM, IN TERMS OF BD-RATE GAIN (%) AND CODING TIME (%).

Res.	Sequence Name	ILR-SQ		ILR-VQ	
		BD-rate	ET/DT	BD-rate	ET/DT
2560	Basketball_Sc	-9.26	105/102	-1.58	209/99
×	MissionCtrlClip2	-10.00	104/101	-1.62	206/101
1440	**Average**	**-9.63**	**105/102**	**-1.6**	**207/100**
	FlyingGraphics	-8.59	107/103	-1.42	225/102
	Desktop	-26.01	103/103	-1.59	220/101
	Console	-33.64	105/103	-1.78	198/102
1920	ChineseEditing	-12.71	101/102	-1.70	234/100
×	MissionCtrlClip3	-9.88	106/105	-1.42	221/100
1080	Robot	-0.32	114/103	-1.08	209/100
	ChinaSpeed	-12.94	107/103	-1.29	107/100
	Average	**-14.87**	**106/103**	**-1.47**	**217/101**
	Web_browsing	-14.75	101/104	-1.67	195/100
1280	Map	-3.42	110/103	-0.89	229/101
×	Programming	-13.98	106/104	-1.46	216/100
720	SlideShow	-14.44	109/103	-1.65	179/98
	SlideEditing	-13.68	104/102	-1.47	229/100
	Average	**-12.05**	**107/103**	**-1.47**	**208/100**
	Total Average	**-13.11**	**105/103**	**-1.49**	**213/100**

version based on Vector Quantization, called ILR-VQ [16]. In this algorithm, instead of Scalar Quantization, a set of QP-dependent codebooks of size 128 are used to code ILR signal. According to Table II, there is a consistent improvement against the JEM5.0.1 with an affordable computational complexity overhead. The lower performance of the ILR-SQ on the Robot sequence is also reflected in a very low block selection rate at all four QPs where less than 1 percent of the blocks are coded by the ILR-SQ. This is due to the low texture complexity of the content.

The RDPCM tool uses a similar prediction scheme to that of ILR-SQ, except that it is applied on residual values [20]–[22]. Therefore, its use was also tested for the performance evaluation of the ILR-SQ, in order to provide a clear view of compression efficiency. For this purpose, "JEM"+"ILR-SQ" was compared against "JEM"+"RDPCM". This resulted in 11% BD-rate gain in favor of ILR-SQ, which means that the correlation that it is capturing by pixel prediction cannot be captured with the residual prediction of RDPCM.

Other tests were also carried out to confirm the efficiency of ILR-SQ. In Random Access (RA) and Low Delay (LD) modes, ILR-SQ was able to improve BD-rate gain by 8.7% and 6.9%, respectively. Moreover, the natural sequences in the JVET CTC were coded by ILR-SQ in AI mode. In this test an average BD-rate gain of 0.2% was achieved.

V. CONCLUSION

An intra prediction algorithm, named In-Loop Residual coding with Scalar Quantization, is proposed to improve the coding efficiency of synthetic video content. In this algorithm, the distant reference inaccuracy problem in case of in-block content change is targeted. To address this problem, the algorithm uses an additional in-loop residual to enable short-distance pixels from inside the block, rather than outside, as references. The proposed technique achieves an average gain of approximately 13% in terms of BD-rate.

REFERENCES

[1] J. Xu, R. Joshi, and R.A. Cohen. Overview of the emerging HEVC screen content coding extension. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(1):50–62, 2016.

[2] X. Xu, S. Liu, T.D. Chuang, Y.W. Huang, S.M. Lei, K. Rapaka, C. Pang, V. Seregin, Y.K. Wang, and M. Karczewicz. Intra block copy in HEVC screen content coding extensions. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 6(4):409–419, 2016.

[3] L. Guo, J. Sole, and M. Karczewicz. Palette mode for screen content coding, document JCTVC-M0323, Incheon. *Korea*, April, 2013.

[4] L. Zhang, X. Xiu, J. Chen, M. Karczewicz, Y. He, Y. Ye, J. Xu, J. Sole, and W.S. Kim. Adaptive color-space transform in HEVC screen content coding. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 6(4):446–459, 2016.

[5] B. Li, J. Xu, G.J. Sullivan, Y. Zhou, and B. Lin. Adaptive motion vector resolution for screen content. In *Proc. 19th Meeting Joint Collaborative Team Video Coding-S0085, Strasbourg, France*, pages 1–14, October, 2014.

[6] M. Wien, V. Baroncini, J. Boyce, A. Segall, and T. Suzuki. Preliminary joint call for evidence on video compression with capability beyond HEVC. In *Document JVET-E1002 5th JVET Meeting, Geneva, Switzerland*, January, 2017.

[7] N. Sidaty, W. Hamidouche, O. Deforges, and P. Philippe. Emerging video coding performance: 4K quality monitoring. In *IEEE conf. on Quality of Multimedia Experience (QoMEX)*, pages 1–3, 2017.

[8] J. Chen, E. Alshina, G.J. Sullivan, J.R. Ohm, and J. Boyce. Algorithm description of joint exploration test model 7 (JEM7). In *Document JVET-G1001 7th JVET Meeting, Torino, Italy*, July, 2017.

[9] Z. Wang, S. Wang, J. Zhang, and S. Ma. Local-constrained quadtree plus binary tree block partition structure for enhanced video coding. In *IEEE conf. on Visual Communications and Image Processing (VCIP), Chengdu, China*, pages 1–4, 2016.

[10] J. An, H. Huang, K. Zhang, Y.W. Huang, and S. Lei. Quadtree plus binary tree structure integration with JEM tools. In *Document JVET-B0023 2nd JVET Meeting, San Diego, USA*, February, 2016.

[11] J. Chen and E Alshina. Algorithm description for versatile video coding and test model 1 (vtm 1). In *Document JVET-J1002 10th JVET Meeting, San Diego, USA*, April, 2018.

[12] A. Alshin and E. Alshina. Bi-directional optical flow for future video codec. In *IEEE International Data Compression Conference (DCC), Snowbird, UT, USA*, pages 83–90, March, 2016.

[13] A. Said, X. Zhao, M. Karczewicz, J. Chen, and F. Zou. Position dependent prediction combination for intra-frame video coding. In *IEEE conf. on Image Processing (ICIP), Phoenix, AZ, USA*, pages 534–538, 2016.

[14] X. Zhao, J. Chen, A. Said, V. Seregin, H.E. Egilmez, and M. Karczewicz. Nsst: Non-separable secondary transforms for next generation video coding. In *IEEE conf. on Picture Coding Symposium (PCS), Nuremberg, Germany*, pages 1–5, 2016.

[15] X. Zhao, J. Chen, M. Karczewicz, L. Zhang, X. Li, and W.J. Chien. Enhanced multiple transform for video coding. In *IEEE conf. on Data Compression Conference (DCC), Snowbird, UT, USA*, pages 73–82, 2016.

[16] M. Abdoli, F. Henry, P. Brault, P. Duhamel, and F. Dufaux. Intra prediction using in-loop residual coding for the post-HEVC standard. In *IEEE 19th International Workshop on Multimedia Signal Processing (MMSP), Luton, United Kingdom*, pages 1–6, October 2017.

[17] Andrea Gabriellini, David Flynn, Marta Mrak, and Thomas Davies. Combined intra-prediction for high-efficiency video coding. *IEEE Journal of selected topics in Signal Processing*, 5(7):1282, 2011.

[18] André Seixas Dias, Saverio G Blasi, Marta Mrak, and Ebroul Izquierdo. Improved combined intra prediction for higher video compression efficiency. In *IEEE conf. on Picture Coding Symposium (PCS), Nuremberg, Germany*, pages 1–5, 2016.

[19] Xiaoran Cao, Changcai Lai, Yunfei Wang, and Yun He. Short distance intra coding scheme for HEVC. In *IEEE conf. on Picture Coding Symposium (PCS), Krakow, Poland*, pages 501–504, 2012.

[20] Y. H. Tan, C. Yeo, and Z. Li. Residual DPCM for lossless coding in HEVC. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2021–2025, May 2013.

[21] Naccari M., Mrak M., Gabriellini A., Blasi S., and Izquierdo E. Inter prediction residual DPCM, document JCTVC-M0442, Incheon. *Korea*, April, 2013.

[22] R. Joshi, J. Sole, and M. Karczewicz. Ahg8: Residual DPCM for visually lossless coding, document JCTVC-M0351, Incheon. *Korea*, April, 2013.

[23] M.J. Weinberger, G. Seroussi, and G. Sapiro. The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS. *IEEE Transactions on Image processing*, 9(8):1309–1324, 2000.

[24] M. Mrak and J. Xu. Improving screen content coding in hevc by transform skipping. In *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, pages 1209–1213, Aug 2012.

[25] Haoping Yu, Robert Cohen, Krishna Rapaka, and Jizheng Xu. Common conditions for screen content coding tests. *Joint Collaborative Team on Video Coding (JCT-VC)*, 2014.

[26] K. Suehring and X. Li. JVET common test conditions and software reference configurations. In *Document JVET-H1010 8th JVET Meeting, Macau, China*, October, 2017.