Intra Block-DPCM With Layer Separation of Screen Content in VVC

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

To cite this version:

Mohsen Abdoli, Felix Henry, Patrice Brault, Frédéric Dufaux, Pierre Duhamel, et al.. Intra Block-DPCM With Layer Separation of Screen Content in VVC. 26th IEEE International Conference on Image Processing (ICIP 2019), Sep 2019, Taipei, Taiwan. 10.1109/icip.2019.8803389 . hal-02116897

HAL Id: hal-02116897
https://hal.science/hal-02116897v1

INTRA BLOCK-DPCM WITH LAYER SEPARATION OF SCREEN CONTENT IN VVC

Mohsen Abdoli*†*, Félix Henry***, Patrice Brault*†*, Frédéric Dufaux*†*, Pierre Duhamel*†*, Pierrick Philippe**

* Orange Labs, Cesson Sévigné, France
†L2S, CNRS - CentraleSupelec - Université Paris-Sud, Gif-sur-Yvette, France

ABSTRACT

An intra coding algorithm with layer separation is proposed. This algorithm is designed on top of an adopted tool in VVC, called Block DPCM (BDPCM), and benefits from texture information in a neighborhood to derive intensity levels of background and foreground layers. This information is used to reduce large rate of residual in case of incorrect layer prediction by BDPCM. For this purpose, three inter-layer transition states are defined that are either implicitly or explicitly conveyed to the decoder. Once a transition is signaled, the decoder corrects the prediction value using the derived layer information. Experiments on screen contents show a BD-rate gain of about 10% percent over VVC Test Model (VTM) and 1% over the regular BDPCM, with the cost of computational complexity.

***Index Terms*—** Intra coding, Versatile Video Coding, Screen Content Coding

1. INTRODUCTION

The next generation video coding standard, called Versatile Video Coding (VVC), is currently under development [1]. The versatility aspect of VVC emphasizes the fact that it is supposedly responsible for coding a wide variety of video formats, including screen content. From this aspect, VVC is in contrast with its predecessor, High Efficiency Video Coding (HEVC), which integrates its dedicated technologies in an extension called HEVC Screen Content Coding (SCC) [2].

Texture coding problem of screen content is essentially different from natural content. This is mainly due to unnatural and sharp edges with less angular diversity in screen contents. Therefore, different tools have been proposed to specifically target this type of content. Some examples are Residual DPCM (RDPCM) [3], Transform Skip Mode (TSM) [4], Palette [5], Intra Block Copy (IBC) [6], Adaptive color space transform [7] and Unary Bitplane Coding (UBC) of screen content residuals [8].

One of the technologies that has been adopted for intra coding of screen content with VVC is Block DPCM (BDPCM) [9–13]. The proposed algorithm is an extension of the regular BDPCM to improve it for bi-layer content, where the transition between background and foreground is sharp. Usually, this type of content can cause a high rate of residual by BDPCM and the proposed layer separation algorithm in this paper aims at addressing this issue.

2. REGULAR BLOCK-DPCM

The main contribution of BCPDM is keeping the full reconstruction at the pixel level [12]. Such independence allows the use of in-block pixels as reference for intra prediction, which is in contrast with the regular intra prediction [14]. In fact, the full reconstruction of an individual pixel by the regular intra prediction depends on all other pixels within the same block. This is due to the residual signal which has to go through the block level steps of transformation and quantization. In contrast, the residual coding in BDPCM does not employ any of these steps.

The prediction process of each pixel with BDPCM consists of four main steps. In a nutshell, these steps predict each pixel using its in-block references, then reconstruct it in order to be used as in-block reference for next pixels in the rest of the block. In this section, these four steps are briefly discussed.

2.1. In-block pixel prediction

BDPCM uses three reference pixels at left (α), above (β) and above-left (γ) for prediction of each pixel. Depending on position of the pixel in the block, these references can fall either outside of the block and among the regular intra references, or inside the block.

A context-adaptive pixel predictor, called LOCO-I, detects possible edges in the references [15]. This predictor has three implicit internal modes for predicting pixel p, as expressed in Eq. 1. The first and second modes of LOCO-I represent vertical and horizontal edges, respectively. Conversely, when no edge is detected, the third mode is used. This predictor function has previously been used for short distance intra coding of VVC [16].

$$p = \begin{cases} min(\alpha, \beta), & \text{if } \gamma \leq max(\alpha, \beta). \\ max(\alpha, \beta), & \text{if } \gamma \geq min(\alpha, \beta). \\ \alpha + \beta - \gamma, & \text{Otherwise.} \end{cases} \quad (1)$$

2.2. Residual calculation

Once the prediction value is calculated, its residual is calculated. Since the residual at this stage is lossless and inaccessible at the decoder side, it is denoted as $\tilde{r}$ and calculated as the subtraction of the original pixel value o from p:

$$\tilde{r} = o - p. \tag{2}$$

2.3. Residual quantization

The pixel-level independence is achieved by skipping the residual transformation and integrating a spatial domain quantization. This is performed by a linear quantizer Q as expressed in Eq. 3. To accommodate the correct rate-distortion ratio, imposed by the Quantizer Parameter (QP), BDPCM adopts the spatial domain normalization used in the TSM method [4]. The quantized residual value r is transmitted by the encoder.

$$r = Q(\tilde{r}). \tag{3}$$

2.4. Pixel reconstruction

The last stage of the regular BDPCM is the pixel reconstruction using p and r, from previous steps, as expressed in Eq. 4. Once reconstructed, current pixel can be used as an in-block reference for other pixels within the same block.

$$c = p + r. \tag{4}$$

3. BDPCM WITH LAYER SEPARATION

The prediction scheme in the regular BDPCM algorithm introduces a relatively large residual, when the original pixel value is far from its prediction. In screen content, this usually happens when in-block references belong to background layer, while current pixel belongs to foreground layer, or vice versa. In this situation, the available information in references is not adequate for an accurate prediction. This problem is called layer transition in the remainder of this paper and a corresponding pixel is called transition pixel.

The proposed contribution to BDPCM algorithm is an information derivation about foreground and background layers from a local neighborhood. This information along with an adaptive transition detection helps at reducing residual energy of transition pixels. Figure 1 shows three bi-layer screen content blocks. As can be seen, in all examples, the local neighborhood around the current block contains useful information about intensity level of the background and foreground layers.

3.1. Implicit layer detection

The proposed method uses texture information of previously coded blocks for foreground and background layer separation. This allows implicit layer detection at the decode side and is in contrast with explicit tools such as Palette mode.

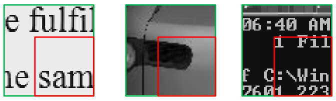


Fig. 1. Bi-layer screen content blocks (red squares) with their neighborhoods (green squares).

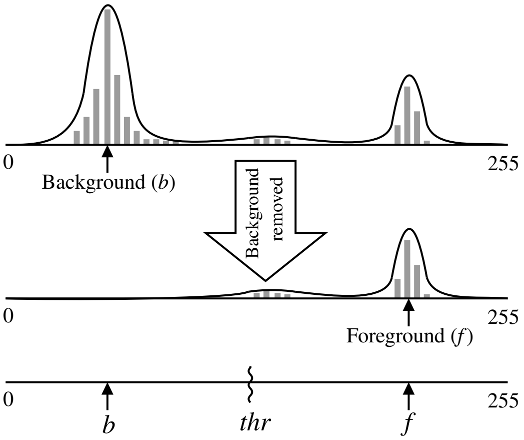
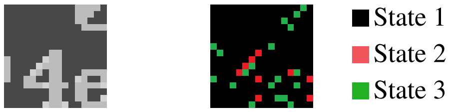


Fig. 2. Two trials of GMM-based background and foreground layer detection from histogram and a transition threshold.

The amount of texture information used from previous blocks is a compromise between complexity and layer separation accuracy. The complexity is defined both in terms of the memory usage of storing previous reconstructed blocks and the computation overhead of processing their pixels. For a more practical implementation, especially at the decoder side, the proposed design simply uses the texture information from above row and left column of the block.

Proposed layer detection uses a simplified Gaussian Mixture Modeling (GMM) on the histogram of texture information. As the conventional GMM parameter estimation methods, such as Expectation-Maximization, are too complex to be performed for each block in video coding, here a low-complexity greedy algorithm is used for this purpose, similar to [17]. Each trial of this algorithm spots one major neighborhood around the highest peak in a histogram as the most popular intensity level and estimates it by a Gaussian distribution. In the current problem, we perform two trials of this algorithm for finding the intensity levels of background (b) and foreground (f) layers, respectively. This process is summarized in Fig. 2. As can be seen, a threshold thr is also defined after determining the intensity levels of layers. This threshold will subsequently be used for inter-layer transition at the encoder side.



a) Original block *b*) Transition state map

Fig. 3. An example 16×16 block with background and foreground layers along with its transition state map.

3.2. Inter-layer transition states

The proposed algorithm improves the context-adaptive predictor of Eq. 1, when it makes an incorrect prediction at an undetectable edge between two layers. In this case an inter-layer transition is applied to correct the prediction value with the intensity level of a proper layer. However, use of the inter-layer transition is limited for pixels with certain condition. Depending on a local neighborhood around a pixel, three states can be defined:

State 1) Plain area which is efficiently handled by the context-adaptive predictor of BDPCM. Hence, no transition is allowed. The decoder can identify this state implicitly.

State 2) Intra-layer low-complex area which is usually characterized by a moderately accurate prediction along with a small residual value. The inter-layer transition is allowed for this condition, but is only applied when the residual error is large enough to cross through the threshold thr. A flag explicitly informs the decoder about the absence of the transition in this state.

State 3) Inter-layer high-complex area on which the context-adaptive predictor performs poorly and produces a large residual. The inter-layer transition is allowed and explicitly signaled to the decoder in this state.

An example of a 16×16 block is presented in Fig. 3, which consists of a bright text on a dark background. The transition state map, at the right of this figure, shows how the above three states are assigned to pixels with different levels of BDPCM prediction error.

3.3. Block coding

In the proposed algorithm, layer affiliation of pixels is conveyed with BDPCM residual information. For simplicity, the decoder side process is explained first and then order of the encoder side process to match the decoder side is provided.

In the rest of this section, the following notation is used. Symbols p, c and o respectively present prediction, reconstruction and original values of each pixel in the block. Moreover, each residual value is associated with its signed decoded residual r which is decomposed into its amplitude a and sign sgn, while its transition state and transition flag are denoted as s and t, respectively. Finally, block level symbols b and f represent intensity levels of background and foreground layers, with a threshold thr, as shown in Fig. 2.

Algorithm 1 The decoder of BDPCM with layer separation.

```
procedure DECODER
    Set background $b$ and foreground $f$ as explained in Sec. 3.1
    for each pixel in block do
        Parsing
        Parse amplitude $a$
        if $a = 0$ then
            $r \leftarrow 0$
            $s \leftarrow 1$ (implicit)
        else
            Parse sign $sgn$
            $r \leftarrow sgn \times a$
            Parse transition flag $t$
            if $t = 0$ then
                $s \leftarrow 2$ (explicit)
            else
                $s \leftarrow 3$ (explicit)
        Decompression
        Predict $p$ using the function in Eq. 1
        if $s = 3$ then
            if $p - b < p - f$ then
                Background to foreground transition
                $p \leftarrow f$
            else
                Foreground to background transition
                $p \leftarrow b$
        Reconstruct $c \leftarrow p + r$
```

The decoder side of the regular BDPCM algorithm includes two phases of residual parsing and block decompression. The parsing phase simply consists of two steps for each residual value: 1) parsing the amplitude a, and 2) parsing the sign sgn if a is non-zero. After the parsing phase, signed residual value r for each pixel is available. Then the decompression phase is performed in two steps for reconstructing each pixel: 1) prediction of p by the context-adaptive function in Eq. 1, and 2) reconstruction of c by adding residual to the prediction value.

The proposed BDPCM with layer separation adds a few extra steps to above phases, as shown in Algorithm 1. In the parsing phase, a combination of explicit and implicit derivation of the transition states is added. More precisely, the following steps are performed parsing each residual value: 1) parsing the amplitude a (similar to the regular BDPCM), 2) parsing the sign sgn if a is non-zero (similar to the regular BDPCM), 3) Implicit state derivation of $s = 1$, if the amplitude is zero, and finally 4) explicit state derivation of $s = 2$ or 3, by parsing the transition flag t, if a is non-zero. Each pixel at this stage has its signed residual value r, state s and transition flag t (if $s \neq 1$).

Once the residual amplitudes, states and transition flags are parsed, the BDPCM decoder is able to perform the decompression phase for block reconstruction. In this phase, first the layer derivation is performed according to Sec. 3.1, in order to obtain background b and foreground f intensity levels for the entire block. Then the following steps are carried out for reconstruction of each pixel: 1) prediction of p by the context-adaptive function of Eq. 1, 2) in the case of being in State 3, making the inter-layer transition. For this purpose, first the distance of p from both b and f is calculated and then p is updated with the intensity level of the layer with longer distance. Finally, 3) pixel c is reconstucted normally and by adding the residual r.

An encoder conforming the above decoder syntax would properly determine residual states and their transition flags. Algorithm 2 summarizes this process. In this algorithm, the regular BDPCM prediction is performed on each pixel first. Then a condition checks whether the p and o fall on different sides of thr, in order to determine necessity of an inter-layer transition. In case that a transition is required (i.e. $s = 3$), the distance of p from both layers b and f is compared. Then p is replaced with the further layer. Otherwise, the significance of amplitude value a is used to decide between implicit or explicit state derivation of $s = 1$ or $s = 2$, respectively. Finally, the residual is calculated, quantized and transmitted with its sign and amplitude separately.

Algorithm 2 The encoder of BDPCM with layer separation.

```
procedure ENCODER
    Set background $b$ and foreground $f$ as explained in Sec. 3.1
    for each prediction pixel in block do
        Obtain $p$ using the function in Eq. 1
        if $p < thr < o$ or $o < thr < p$ then
            $s \leftarrow 3$
            $t \leftarrow 1$
            Encode $t$ (explicit)
            if $p - b < p - f$ then
                Background to foreground transition
                $p \leftarrow f$
            else
                Foreground to background transition
                $p \leftarrow b$
            Obtain $r$ with Eq. 2 and Eq. 3
            Decompose $r$ in$sgn$ and $a$
            Encode $sgn$ and $a$
        else
            Obtain $r$ with Eq. 2 and Eq. 3
            Decompose $r$ in$sgn$ and $a$
            Encode $sgn$ and $a$
            if $r = 0$ then
                $s \leftarrow 1$ (implicit)
            else
                $s \leftarrow 2$
                $t \leftarrow 0$
                Encode $t$ (explicit)
```

Table 1. Performance of the proposed BDPCM with layer separation against VTM and regular BDPCM, in terms of BD-rate gain (%) and coding time (%).

Res.	Sequence	vs. VTM		vs. VTM+BDPCM	
		BD-rate	ET/DT	BD-rate	ET/DT
2560 × 1440	Basketball_Sc	-8.95	135/142	-1.28	139/139
	MissionCtrlClip2	-7.60	135/148	-1.35	144/142
	Average	**-8.27**	**135/145**	**-1.31**	**142/141**
1920 × 1080	FlyingGraphics	-7.21	149/153	-2.59	144/149
	Desktop	-20.42	150/127	-0.41	140/130
	Console	-19.06	145/133	-1.25	130/125
	ChineseEditing	-9.94	141/142	-0.94	149/129
	MissionCtrlClip3	-8.92	136/110	-0.24	126/123
	Robot	-4,34	144/133	-1.96	139/120
	ChinaSpeed	-9.42	137/136	-2.09	130/146
	TencentAOV7	-1.49	127/106	-0.28	119/112
	Average	**-6.58**	**141/130**	**-1.22**	**134/129**
1280 × 720	Web_browsing	-14.01	131/115	-0.08	142/126
	Map	-4.15	146/139	-1.26	139/127
	Programming	-11.91	159/144	-2.50	150/162
	SlideShow	-12.36	141/117	-0.52	142/121
	SlideEditing	-12.26	134/142	-1.17	129/130
	BasketballDrillText	-1.05	114/102	-0.00	113/109
	Average	**-9.29**	**137/126**	**-0.92**	**135/129**
	Total Average	**-9.56**	**139/131**	**-1.05**	**144/130**

4. RESULTS

The performance evaluation of the proposed BDPCM algorithm with layer separation is carried out within the VVC Test Model (VTM-1.0) reference software [18]. For this purpose, the algorithm has been implemented on top of VTM+BDPCM and its performance has been calculated against two anchor encoders: 1) reference VTM, and 2) VTM+BDPCM. Table 1 compares the performance of the proposed algorithm against these two anchor encoders, under the constraints of the Common Test Conditions (CTC) [19] with screen content from HEVC-SCC. As can be seen, the proposed method improves the VTM by about 10%, in terms of BD-Rate gain [20], while the encoder and decoder complexity increases by 44% and 30% respectively. The results in Table 1 also confirms that the proposed layer can improve the regular BDPCM. As can be seen, an average BD-Rate gain of about 1% is achieved on top of the regular BDPCM.

5. CONCLUSION

An intra coding method is proposed to integrate a layer separation step on top of the existing BDPCM algorithm in VVC. This method improves the coding efficiency of BDPCM when it is applied on content with sharp edges background and foreground layers. As of today, BDPCM has been adopted in VVC and may still evolve until the end of the standardization process. Therefore, one can adapt the proposed idea of this paper to the future versions of BDPCM.

6. REFERENCES

[1] Sullivan G. Video coding standards progress report: Joint video experts team launches the Versatile Video Coding project. *SMPTE Motion Imaging Journal*, 127(8):94–98, 2018.

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

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

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

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

[6] Xu X., Liu S., Chuang T., Huang Y., Lei S., Rapaka K., Pang C., Seregin V., Wang Y., and Karczewicz M. 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.

[7] Zhang L., Xiu X., Chen J., Karczewicz M., He Y., Ye Y., Xu J., Sole J., and Kim W. 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.

[8] Abdoli M., Henry F., Brault P., Dufaux F., and Duhamel P. Transform coefficient coding for screen content in Versatile Video Coding (VVC). In *International Conference on Acoustics, Speech, and Signal Processing, Brighton, UK*, pages 1792–1796, 2019.

[9] Abdoli M., Henry F., Philippe P., , and Clare G. BDPCM with horizontal/vertical predictor and independently decodable areas. *Document JVET-M0057*, Marrakech, Morocco, January 2019.

[10] Karczewicz M. and Coban M. Quantized residual BDPCM. *Document JVET-N0416*, Geneva, Switzerland, March 2019.

[11] Clare G., Henry F., Bross B., Nguyen T., Keydel P., Schwarz H., Marpe D., Wiegand T., Zhao X., Li X., Xu X., and Liu S. BDPCM with harmonized residual coding and CCB limitation. *Document JVET-N0214*, Geneva, Switzerland, March 2019.

[12] Abdoli M., Henry F., Brault P., Duhamel P., and Dufaux F. Short distance intra prediction of screen content in versatile video coding (VVC). *IEEE Signal Processing Letters*, 24(5):564–568, 2018.

[13] Abdoli M., Clare M., Henry F., and Philippe P. AHG11: Block DPCM for screen content coding. *Document JVET-L0078*, Macao, China, October 2018.

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

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

[16] Abdoli M., Henry F., Brault P., Duhamel P., and Dufaux F. 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, 2017.

[17] Abdoli M., Sarikhani H., Ghanbari M., and Brault P. Gaussian mixture model-based contrast enhancement. *IET image processing*, 9(7):569–577, 2015.

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

[19] Yu H., Cohen R., Rapaka K., and Xu J. Common test conditions for screen content coding. In *Document JCTVC-X1015, Geneva, Switzerland*, January, 2015.

[20] Bjontegaard G. Calculation of average PSNR differences between rd-curves. In *Document VCEG-M33, Austin, Texas, USA*, April, 2001.