MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Independent Uniform Prediction Mode for Screen Content Video Coding

Zhang, X.; Cohen, R.A.; Vetro, A.

TR2014-105 December 2014

Abstract

Many of the existing video coding standards in use today were developed primarily using camera-captured content as test material. Today, with the more widespread use of connected devices, there is an increased interest in developing video coding tools that target screen content video. Screen content video is often characterized by having sharp edges, noiseless graphics-generated region, repeated patterns, limited sets of colors, etc. This paper presents an independent uniform prediction (IUP) mode for improving the coding efficiency of screen content video. IUP chooses one color out of a small set of global colors to form a uniform prediction block. Unlike existing palette-based modes, IUP does not have to construct and signal a color index map for every block that is coded. Experimental results using IUP in the HEVC Range Extensions 6.0 framework are presented, along with results using techniques that reduce complexity so that the IUP-based encoder is faster than the reference encoder.

IEEE Visual Communications and Image Processing (VCIP 2014)

Independent Uniform Prediction Mode for Screen Content Video Coding

Xingyu Zhang [1], Robert A. Cohen [2], Anthony Vetro [2]

[1] *The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong*
[2] *Mitsubishi Electric Research Laboratories 201 Broadway, Cambridge, MA 02139*
eexyzhang@ust.hk, {cohen, avetro}@merl.com

***Abstract*—Many of the existing video coding standards in use today were developed primarily using camera-captured content as test material. Today, with the more widespread use of connected devices, there is an increased interest in developing video coding tools that target screen content video. Screen content video is often characterized by having sharp edges, noiseless graphics-generated region, repeated patterns, limited sets of colors, etc. This paper presents an independent uniform prediction (IUP) mode for improving the coding efficiency of screen content video. IUP chooses one color out of a small set of global colors to form a uniform prediction block. Unlike existing palette-based modes, IUP does not have to construct and signal a color index map for every block that is coded. Experimental results using IUP in the HEVC Range Extensions 6.0 framework are presented, along with results using techniques that reduce complexity so that the IUP-based encoder is faster than the reference encoder.**

***Index Terms*—Screen content coding, HEVC, prediction modes, independent uniform prediction**

I. Introduction

With the recent proliferation of connected multimedia devices, screen content video is becoming ubiquitous in many applications, e.g. screen sharing, remote desktop, online education, mixed graphics and camera-captured programming, etc. State-of-the-art video coding standards like H.264/AVC and H.265/HEVC [1] provide good video compression performance. These standards, however, were developed primarily with camera-captured content in mind, whereas screen content video has many distinguishable features not present in camera-captured video. For example, screen content typically contains more sharp edges, regions of identical pixels values with no noise, repeated patterns, and a limited set of unique colors, e.g. as defined by a color theme. By exploiting these unique features of screen content, the compression performance of these standards can be improved.

Recently, some screen content coding tools have been developed for extensions of HEVC. For example, an intra block copy (IntraBC) mode is proposed in [2] for 2Nx2N blocks to reduce spatial redundancy in an intra-coded picture. Analogous to motion estimation, a displacement vector is used to identify a previously coded block in the current picture that can be used as a prediction for the current block. Refinements to IntraBC are presented in [3], where the IntraBC mode is extended to finer block structures, i.e. NxN, 2NxN and Nx2N. Examples of palette or colormap-based approaches are described in [4], [5]. In palette mode, a limited set of colors is identified, and then an index map of colors and the color values are signaled at a block level. The palette is updated as blocks are being coded, so there is some local adaptation of the colormap table. Several refinements of palette mode are being studied in [6]. If a color in the current block, however, needs to be added to the palette, then additional overhead is needed to signal that color value. Also, some distortion in the prediction may occur because a given color in the current block may be quantized to a different color. Additionally, the local adaptation of the palette may cause some commonly-used colors to be repeatedly removed and then reinserted, creating additional signaling overhead.

978-1-4799-6139-9/14/$31.00 ©2014 IEEE

In this paper, we propose an independent uniform prediction (IUP) mode for both intra- and inter-coded blocks to exploit a set of globally occurring colors in a picture. Such colors are commonly found in computing environments or in videos containing computer graphics and text, for example, as a pair of foreground and background colors. Flat blocks containing only these foreground or background colors are frequently found in computer-generated videos. With IUP, a small set of global colors is obtained by either pre-analysis of the picture content or by having a set of pre-defined colors, such as the foreground and background specified by the color scheme being used by the device generating the video. At the coding unit (CU) level, IUP chooses one of the global colors to form a uniform prediction block and then signals an index to indicate the selected color. Unlike existing palette-based implementations, IUP does not have to construct and signal a color index map for every CU. Moreover, all the pixels inside the CU can be predicted simultaneously.

The rest of this paper is organized of follows. In Section II, we introduce IUP mode and discuss its use for coding screen content. In Section III, we describe how IUP is implemented in the encoder. Experimental results for different implementations of IUP are presented in Section IV. Finally, Section V concludes the paper.

II. Independent Uniform Prediction Mode

In the following discussion, we use the RGB 444 color space for illustration. Similar analysis or calculations can be easily extended to sequences having other formats. Section IV includes experimental results for both RGB and YCbCr videos.

To use IUP mode, a subset of major colors is defined. These colors are identified prior to coding a slice. Each of the N major colors in a slice is denoted as a color triplet $S^i : \{r^i, g^i, b^i\}, \ i \in [1, N]$. We signal the count N and the set of color triplets S^i in the slice header. There are many ways to select the major colors. For example, in a screen sharing application, the computer system knows what the common background or foreground colors are, so those could comprise the set of major colors. For scenarios in which the encoding is done separately from the content generation, the major colors can be selected directly from the video content. For example, since we are looking for a global set of major colors, we can sub-sample the slice and then calculate the histogram of all the triplets in the sub-sampled slice. The N most frequently used triplets are selected as candidate color triplets to form the prediction block at a CU level.

At the CU level, IUP is available as an additional prediction mode. One of the candidate color triplets S^* is selected to form a uniform-colored prediction block

$$p_r(k,l) = r^*, \quad p_g(k,l) = g^*, \quad p_b(k,l) = b^*, \qquad (1)$$

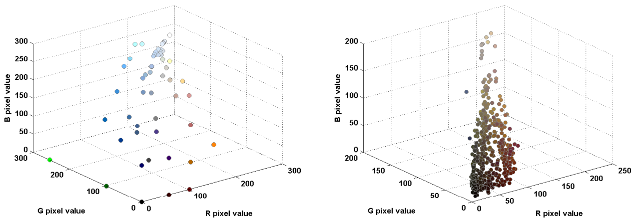
where (k, l) are the coordinates of the pixel within the CU. In HEVC, the intra prediction process of an intra CU operates at the Transform Unit (TU) level. Although every TU inside the Prediction Unit shares the same prediction mode, the prediction process is performed sequentially depending on the reconstructed pixels in the previous TUs. However, in IUP mode there is no need to perform the prediction process sequentially at the TU level, because the prediction is independent of the neighboring blocks. Thus, for a given color triplet, the prediction block only needs to be computed once for each CU. The index of the triplet S^* used to predict a CU is signaled at the CU level.

Note that the existing DC prediction mode also generates mostly uniform-colored prediction blocks. The colors in this prediction, however, are dictated by neighboring pixels from adjacent previously decoded blocks. For screen content video, however, the CU quadtree architecture frequently results in flat CUs being adjacent to textured CUs. The colors computed with the DC prediction mode are averaged or filtered values of the neighboring pixels. These values may therefore not perfectly match the color of a flat or mostly flat block, resulting in an increased prediction error. The set of major colors available with IUP mode increases the opportunity for a uniform prediction to have a small or zero prediction error.

The number of major colors N does not necessarily need to be the same for all slices or over different sequences. In our design, in addition to identifying the most frequently occurring colors from the histogram, we also consider whether these frequencies are significant enough to deem the corresponding colors as major colors. This significance is determined by applying a threshold T_1 to the frequencies. Camera-captured content typically contains a large number of different colors, whereas screen content contains much fewer unique colors.

(a) *Desktop* (b) *EBURainFruits*

Fig. 1. Snapshots of a screen content sequence and a natural content sequence



(a) *Desktop: 53 unique colors* (b) *EBURainFruits: 506 unique colors*

Fig. 2. Visualization of unique RGB colors, for down-sampled pictures

This behavior can be illustrated for the pictures shown in Fig. 1. *Desktop* is screen content material, and *EBURainFruits* is an example of camera-captured content. Both sequences have a picture resolution of 1920×1080. These pictures are down-sampled by 64:1 in both the horizontal and vertical dimensions, without filtering, and then the sets of unique colors present in the down-sampled pictures are shown in the corresponding plots of Fig 2. We can see that there are significantly fewer unique colors in Fig. 2a than in Fig. 2b. In camera-captured pictures, there are rarely totally flat regions, i.e. regions where all pixels have exactly the same value, due to camera noise and lighting conditions. Thus, the chance for just one or two color triplets to occur at a very high frequency is low. This characteristic results in there being $N = 0$ major colors, given a properly chosen threshold T_1. When $N = 0$, IUP mode is not considered for coding any CU in the slice, which means that no IUP-related overhead needs to be signaled for the CU. If $N = 1$, then no overhead is needed to signal which major color to use for a CU using IUP mode.

III. Implementation at the Encoder

In the encoder, rate-distortion-optimization (RDO) can be used to select which of the N major colors to use for IUP. The selected triplet S^* is

$$S^* = arg \ \min_{S^i} D(S^i) + \lambda R(S^i), \qquad (2)$$

where $D(S^i)$ is the distortion term of the reconstructed CU using color triplet S^i under IUP mode. $R(S^i)$ is the number of bits needed to encode the prediction residue block and overhead for the mode. The scaling factor λ is identical to that used for existing modes. The RDO process is also used to choose among other available prediction modes such as IUP, directional intra prediction, IntraBC, and inter prediction. Computing the rate-distortion cost for every available major color, however, can be quite compute-intensive, due to the subsequent transform, quantization, and entropy coding needed to compute distortion and rate metrics. In this section, we describe several techniques that allow us to significantly reduce the complexity of RDO computations for IUP while maintaining the associated coding gains.

In IUP mode, the prediction is a uniform block. The residual block is formed by subtracting the original block from this uniform block. The residual block is then transformed and quantized. If there are more than one color candidates for IUP, the only difference between the corresponding transformed residuals is in the value of their DC transform coefficients. The color candidate that is closest to the mean or DC coefficient value of the CU being coded is therefore more likely to be the optimal IUP color, given that it minimizes the magnitude of the DC coefficient. We can use this concept so that only one of the candidate triplets used in the RDO process. This triplet S^* is selected as follows:

$$S^* = arg \ \min_{S^i} |g^i - \bar{x}_g|, \ i \in [1, N] \tag{3}$$

where $\bar{x}_g$ is the mean of the green component of the current CU. We also tested minimizing the absolute differences for all three components $|g^i - \bar{x}_g| + |b^i - \bar{x}_b| + |r^i - \bar{x}_r|$, but the change in coding performance was trivial (less than 0.01%).

Although the final selected IUP color minimizes the difference $|g^i - \bar{x}_g|$, if this different is too large, IUP may not outperform the other prediction modes. Therefore, if this difference is greater than a threshold T_2, we disqualify IUP for the current CU and proceed to testing other prediction modes in the RDO process.

We also reduce the number of blocks tested with IUP by using variance as a measure of texture complexity. If the texture in a large CU is significantly complex, then the CU is likely to be split into smaller CUs, especially for intra-coded blocks. Since IUP is not designed to predict the high frequency components of a block, we can skip the IUP test for CUs whose variance exceeds a threshold. The threshold is dependent upon the CU size.

For this paper, we modified the syntax of HEVC Range Extensions Draft 6 [7] to support IUP. In the CU header, a flag *IUP_pred_flag* is signaled to indicate whether IUP mode is used. If IUP is used, then a color index *IUP_color_Idx* is signaled to indicate the selected color triplet, if more than one triplet is available. This data is signaled after the *cu_skip_flag* but before the syntax associated with other intra and inter prediction modes. If all pixels in a CU have the same color, and if that color matches the IUP color, then the prediction error for the entire CU will be zero. In this case, we select IUP mode for this CU and skip any tests of other prediction modes by terminating the RDO process early.

IV. Experimental results

The HM13.0-RExt6.0 [8] reference software is modified to incorporate IUP, and the unmodified software is used to generate anchors for comparison. The tests follow the common test conditions described in [9]. Also, IntraBC mode for 2Nx2N, 2NxN, Nx2N and NxN PUs is enabled for both the anchors and tested simulations. The set of 12 RGB and 12 YCbCr sequences include graphics and text, animation, and mixed screen and camera-captured content. Due to the limited space, we show only the results under All Intra (AI) test conditions. Average changes in BD-rate [10] for videos coded using both the full RDO and fast RDO implementations described in Section III are shown in Table I. The performance on individual RGB sequences is shown in Table II. For these experiments, we use up to $N = 2$ major color triplets for each slice. The histogram threshold T_1 for the major color triplets is set to 0.05 empirically, and the threshold T_2 for the mean difference is set to 128. We found in most cases that two major color triplets are ultimately chosen at the slice level for text, graphics, and mixed-content material, and no major color triplets are chosen for animation material. In ancillary experiments we also found, as anticipated, that IUP mode is not exercised with camera-captured content. The encoding times using fast RDO mode with IUP are less than those of the unmodified encoder. The decoding times are reduced in all cases as well, as computing the uniform IUP prediction is less complex than for the other prediction modes. There was no improvement for the *Robot* sequence because this animation sequence was designed to be more like a camera-captured sequence rather than a limited-color computer desktop sequence.

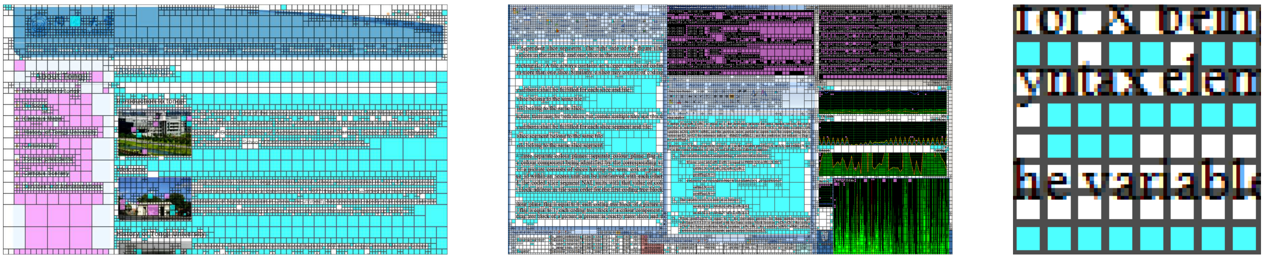
Examples of usage maps for pictures coded with IUP mode enabled are shown in Fig. 3. Here, the CUs are outlined in

TABLE I
Component-wise BD-rate (%) under AI conditions

Class	Full-RDO IUP G/Y	Full-RDO IUP B/Cb	Full-RDO IUP R/Cr	Fast IUP G/Y	Fast IUP B/Cb	Fast IUP R/Cr
RGB, text & graphics,1080p	−2.1	−2.3	−2.3	−2.0	−2.1	−2.2
RGB, text & graphics,720p	−1.9	−2.1	−2.1	−1.7	−1.8	−1.8
RGB, mixed content,1440p	−1.1	−1.1	−1.1	−1.0	−0.9	−1.0
RGB, mixed content,1080p	−0.7	−0.8	−0.8	−0.6	−0.7	−0.7
RGB, Animation,720p	0.0	0.0	0.0	0.0	0.0	0.0
YUV, text & graphics,1080p	−1.7	−2.2	−2.1	−1.8	−2.2	−2.2
YUV, text & graphics,720p	−2.0	−2.3	−3.0	−1.7	−2.1	−2.5
YUV, mixed content,1440p	−0.8	−1.1	−1.1	−0.8	−1.1	−1.0
YUV, mixed content,1080p	−0.6	−1.5	−1.3	−0.6	−1.3	−1.2
YUV, Animation,720p	0.0	0.0	0.0	0.0	0.0	0.0
Encoding Time Ratio	131%			94%		
Decoding Time Ratio	96%			96%		

TABLE II
BD-rate (%) of RGB sequences under AI Conditions

Class	Sequence	Full-RDO IUP G	Full-RDO IUP B	Full-RDO IUP R	Fast IUP G	Fast IUP B	Fast IUP R
1080p t&g	Flyinggraphicstext	−1.6	−1.8	−1.8	−1.6	−1.7	−1.7
	Desktop	−1.8	−1.9	−1.9	−1.6	−1.7	−1.7
	Console	−3.0	−3.1	−3.4	−2.7	−2.8	−3.1
720p t&g	WebBrowsing	−3.5	−4.4	−3.9	−3.0	−3.6	−3.5
	Map	−1.9	−2.0	−2.2	−1.5	−1.5	−1.6
	Programming	−1.1	−1.1	−1.1	−1.0	−1.0	−1.0
	SlideShow	−1.2	−1.2	−1.2	−1.1	−1.1	−1.2
1440p mixed	BasketballScreen	−1.4	−1.3	−1.4	−1.4	−1.3	−1.3
	MissionControlClip2	−0.8	−0.9	−0.9	−0.6	−0.6	−0.7
1080p mixed	MissionControlClip3	−0.9	−1.0	−1.0	−0.8	−0.7	−0.8
	SocialNetworkMap	−0.5	−0.6	−0.5	−0.5	−0.6	−0.5
720p	Robot (anim.)	0.0	0.0	0.0	0.0	0.0	0.0



(a) *WebBrowsing map* (b) *Desktop map* (c) *Magnified Desktop map*

Fig. 3. Usage map of IUP mode for the 1st frame (slice) encoded at QP32

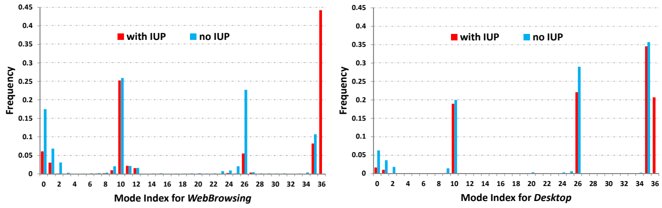
black. Blocks tinted bright blue or pink indicate CUs that use IUP mode. In Fig. 3a blocks tinted blue use (254,255,254) as the (R,G,B) major color triplet in IUP mode, and blocks tinted pink use (238,248,254). These colors correspond to the white background on the right portion of the picture and the light gray/blue background on the left side. In Fig. 3b, blue blocks indicate IUP blocks using (255,255,255), and pink blocks represent IUP blocks that use (0,0,0). A magnified portion of Fig. 3b is shown in Fig. 3c. We can see that IUP mode is often selected for flat blocks that are adjacent to non-flat blocks. This behavior is expected, because with other modes the non-flat blocks generate a prediction block that is either non-flat, or the flat prediction is not a close match due to averaging across boundary pixels having different colors or colors affected by quantization error. For IntraBC mode, there is either no good matching block within the IntraBC search range, or the cost for signaling IUP is less than that of IntraBC.

To get an idea of what prediction modes are being displaced by IUP, we show in Fig. 4 histograms of the final prediction modes for 16x16 CUs from *WebBrowsing* and *Desktop* under all-Intra conditions with QP=32, both with and without IUP enabled. Mode index 0 represents planar mode. Mode 1 is DC mode. Modes 2-34 are the directional prediction modes. Mode 35 is IntraBC, and Mode 36 is IUP. We can see that when IUP is used, the usage frequencies of other prediction modes decrease, especially for planar mode 0 and vertical mode 26. In the reference software, the planar and vertical prediction modes are often present in the list of most probable modes (MPM) for intra prediction, as they are the default modes inserted into the MPM set when space is available. We analyzed the both the rate and distortion contributions to the RD cost function when IUP mode was selected. IUP was frequently selected because it either yielded significantly lower distortion than the other modes, or many modes yielded zero distortion but IUP had the lowest rate.

An additional experiment was performed to determine the gain of IUP over a reference that also includes the palette mode. The improvements in BD-Rate for this case are up to 1.6%.

V. Conclusions

In this paper, we presented an efficient independent uniform prediction (IUP) mode for coding screen content video. This mode leveraged the paradigm that screen content typically includes some common globally-used colors, such as foreground



(a) *WebBrowsing* (b) *Desktop*

Fig. 4. Histograms of prediction modes for 16x16 CUs

or background colors. IUP mode used a set of slice-level colors to compute a prediction block that targeted primarily flat blocks where most pixels have the same color. We also developed several techniques for using IUP in an encoder without increasing its run time, while achieving almost the same improvements in coding efficiency as the straightforward implementation using full rate-distortion optimization. BD-Rate gains of up to 4.4% for RGB components were achieved, with overall average gains of up to 2.3% with full RDO and 2.2% with the fast implementation.

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. on Circuits Systems for Video Technology, Vol. 22, No. 12, pp. 1649-1668, Dec. 2012.

[2] C. Pang, J. Sole, L. Guo, M. Karczewicz and R. Joshi, "Non-RCE3: Intra Motion Compensation with 2-D MVs," ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), JTVC-N0256, Vienna, Austria, Aug. 2013.

[3] T. S. Chang, et al., "RCE3: Results of Subtest D.2 on Nx2N/2NxN/NxN Intra Block Copy," ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), JCTVC-P0180, San Jose, USA, Jan. 2014.

[4] C. Lan, G. Shi and F. Wu, "Compress Compound Images in H.264/MPEG-4 AVC by Exploiting Spatial Correlation," IEEE Trans. on Image Processing, Vol. 19, No. 4, pp. 946-957, April. 2010.

[5] Z. Pan, et al., "A low-complexity screen compression scheme for interactive screen sharing," IEEE Trans. on Circuits Systems for Video Technology, Vol. 23, No. 6, pp. 949-960, Jun. 2013.

[6] Y. W. Huang, et al., "Description of Screen Content Core Experiment 3 (SCCE3): Palette mode," ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), JCTVC-Q1123, Valencia, Spain, Apr. 2014.

[7] D. Flynn, et al., "HEVC Range Extensions Draft 6," ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), JCTVC-P1005, San Jose, USA, Jan. 2014.

[8] HM-13.0+RExt-6.0 software, available online https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-13.0+RExt-6.0/

[9] ITU-T Q6/16 Visual Coding and ISO/IEC JTC1/SC29/WG11 Coding of Moving Pictures and Audio, "Joint Call for Proposals for Coding of Screen Content," ISO/IEC JTC1/SC29/WG11 MPEG2014/N14175, San Jose, USA, Jan. 2014

[10] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," ITU-T Video Coding Experts Group, VCEG-M33, Apr. 2001.