

Lossless 4:2:0 Screen Content Coding Using Luma-Guided Soft Context Formation

Hannah Och and André Kaup
Friedrich-Alexander-Universität Erlangen-Nürnberg
Multimedia Communications and Signal Processing
Cauerstraße 7, 91058 Erlangen, Germany

Abstract—The soft context formation coder is a pixel-wise state-of-the-art lossless screen content coder using pattern matching and color palette coding in combination with arithmetic coding. It achieves excellent compression performance on screen content images in RGB 4:4:4 format with few distinct colors. In contrast to many other lossless compression methods, it codes entire color pixels at once, i.e., all color components of one pixel are coded together. Consequently, it does not natively support image formats with downsampled chroma, such as YCbCr 4:2:0, which is an often used chroma format in video compression. In this paper, we extend the soft context formation coding capabilities to 4:2:0 image compression, by successively coding Y and CbCr planes based on an analysis of normalized mutual information between image planes. Additionally, we propose an enhancement to the chroma prediction based on the luminance plane. Furthermore, we propose to transmit side-information about occurring luma-chroma combinations to improve chroma probability distribution modelling. Averaged over a large screen content image dataset, our proposed method outperforms HEVC-SCC, with HEVC-SCC needing 5.66% more bitrate compared to our method.

Index Terms—lossless coding, screen content, soft context formation, 4:2:0 coding.

I. INTRODUCTION

In contrast to sensor-generated content, screen content (SC) refers to images and videos directly generated by the computer, such as web pages, presentations or graphic user interfaces of applications. In times of increasing home office usage, online conferencing and online learning opportunities, the importance of efficient compression of such data rises. Screen content has unique characteristics which differ from those of sensor-generated data. In general, SC contains only few unique colors, with large uniform areas, sharp contrasts and repetitive image areas, such as letters in textual areas. Compression methods for SC aim to exploit these characteristics. Recent coding standards, such as High Efficiency Video Coding (HEVC) with its SC coding extension [1,2] or Versatile Video Coding (VVC) [3], incorporate coding tools based on these characteristics to improve compression efficiency when coding SC data, such as intra block copy (IBC) or palette mode [2,4].

Since distortions in synthetic image areas are more disturbing to the human visual system than distortions in natural image regions, lossless compression of SC is of high importance [5]. For SC data in RGB 4:4:4 format, there exist multiple dedicated compression techniques, such as the Free Lossless Image Format (FLIF) [6] or JPEG-LS [7], which are based on

pixel-wise prediction followed by entropy coding. Both recent standards HEVC and VVC are capable of lossless coding of SC in RGB 4:4:4 as well [2,4]. All of these methods, however, are outperformed in terms of compression efficiency by soft context formation (SCF) coding [8,9], which employs pixel-wise context-based probability distribution estimation for color triplets in combination with an arithmetic coder.

Although SC data is usually captured in a 4:4:4 format, image and especially video data is often transmitted in YCbCr 4:2:0 format instead, i.e., the chroma planes are downsampled by a factor of two horizontally and vertically. Both video coding standards, HEVC and VVC, are capable of compressing YCbCr 4:2:0 data without loss, ensuring best possible reconstruction quality given the input data. Generally, they code each component separately, but may make use information from the Y plane to better compress the chroma planes, such as chroma block vectors derived from luma block vectors in IBC [10] or chroma sample prediction based on luma samples [11,12]. The pixel-wise lossless compression methods do not natively support 4:2:0 formats and instead workarounds, such as separate coding of each plane, have to be employed. However, in [9], it has been shown for RGB 4:4:4 format that partial lossless coding of intra-frames using SCF can be useful for VVC even for lossy coding. For YCbCr 4:2:0 data, however, this is currently not feasible, since SCF and similar lossless coding approaches do not support the 4:2:0 chroma format. Since SCF coding has proven superior to other methods such as FLIF or JPEG-LS for RGB 4:4:4 SC data [8,9], in this paper, we focus on extending it towards 4:2:0 format capability ensuring more versatility for SCF coding and its possible use cases.

The SCF algorithm codes an image pixel by pixel, where each color is represented as a color triplet containing all color components, e.g., red (R), green (G) and blue (B). For each pixel, the entire color triplet is coded at once. Since 4:2:0 images do not contain all three color components for each pixel, the algorithm cannot be directly extended towards 4:2:0 data. Consequently, this paper is organized as follows: First, we will shortly recap the main points of SCF coding necessary for the following explanations. Next, we explain our coding strategy for extending SCF coding towards 4:2:0 capability by separate luma and chroma coding. We describe our proposed luma-guided enhanced chroma prediction as well as the novel luma-based chroma range coding (CRC)

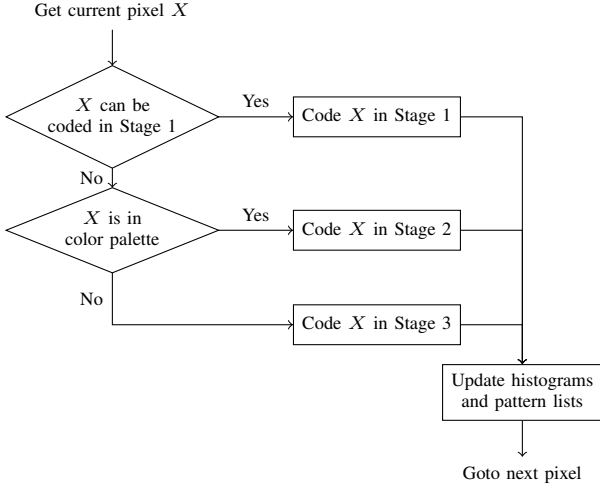


Fig. 1. Overview of the SCF algorithm for one pixel.

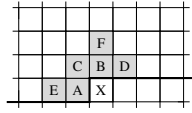


Fig. 2. SCF context $P = \{A, B, C, D, E, F\}$ and the corresponding current pixel at position X .

for improved chroma probability modelling. Finally, we will evaluate the effectiveness of our proposed coding algorithm by comparing the results of our method against VVC and HEVC lossless SC coding.

II. REVIEW OF THE SOFT CONTEXT FORMATION CODER

SCF [8] compresses an image pixel by pixel in raster scan order. During compression, it gathers data about the image to estimate probability distributions for pixel colors. Given these probability distributions, the pixel colors are coded using an arithmetic coder. SCF utilizes three different methods to code pixel colors: A context-based coding in Stage 1, a color-palette-based coding in Stage 2 and residual coding in Stage 3, as depicted in Figure 1. For context-based coding, the SCF algorithm saves all previously seen patterns $P = \{A, B, C, D, E, F\}$ visualized in Figure 2 in a list, and keeps an up-to-date histogram of associated colors at pixel position $X = (m, n)$. Additionally, each pattern histogram contains a count of an escape symbol, representing the number of times the histogram does not contain the color of the current pixel X . To code the current pixel X in Stage 1, all histograms of similar patterns to the current one are merged. The probability for a color $\mathbf{c} = [c_R, c_G, c_B]^T$ is then estimated as

$$p(\mathbf{c}) \approx \frac{N(\mathbf{c})}{N_P}, \quad (1)$$

where $N(\mathbf{c})$ is the count of the color \mathbf{c} in the merged histogram and N_P is the total count of all entries in the histogram. Given an arithmetic coder, we can transmit the color, needing approximately

$$I(\mathbf{c}) \approx -\log_2(p(\mathbf{c})) [\text{bit}]. \quad (2)$$

TABLE I
NORMALIZED MUTUAL INFORMATION BETWEEN IMAGE CHANNELS
AVERAGED OVER DATA SETS.

Data set	RGB 4:4:4			YCbCr 4:2:0		
	R-G	R-B	G-B	Y-Cb	Y-Cr	Cb-Cr
Kodak [15]	0.42	0.31	0.38	0.16	0.14	0.23
SCID [16]	0.85	0.73	0.88	0.32	0.31	0.50

If the current color is not contained in the merged histogram, use of palette-based coding is indicated by transmitting the escape symbol instead.

For Stage 2 coding, the occurrences of each color not coded in Stage 1 are accumulated in a color palette. First, a flag indicates whether the current color is part of the palette. If so, the color of the current pixel is predicted using a modified median adaptive predictor (MAP) [8]. As described in [8], the color palette is split in two sub-palettes. One of them contains all colors similar to the predicted value, the other the rest. A flag indicates which sub-palette is used. The color is then transmitted based on the distribution described by the counts of the chosen sub-palette.

In case the current color is unknown, i.e., not yet included in the color palette, the color prediction error between actual color and MAP predicted color is transmitted per component using a histogram of previously seen prediction errors. Enhancements to this algorithm as described in [9,13] are also applied, in addition to an image adaptive rotation [14] and color transform [8], as well as histogram compaction.

III. EXTENSION OF SOFT CONTEXT FORMATION CODING FOR 4:2:0 COMPRESSION

Since SCF codes entire color triplets at once, it does not natively support SC data in 4:2:0 format. Accordingly, in the following section, we will explain how to effectively extend 4:4:4 based SCF coding towards non-4:4:4 formats.

A. Separate Luma and Chroma Coding

YCbCr is generally assumed to reduce cross-channel dependencies for natural images. In Table I, we evaluate the cross-channel dependencies for natural and SC images for both RGB and YCbCr 4:2:0. We calculate the normalized mutual information (NMI) between R-G, B-G and R-B, as well as Y-Cb, Y-Cr and Cb-Cr channels where NMI is defined as

$$\text{NMI}(X_1; X_2) = \frac{2(H(X_1) - H(X_1|X_2))}{H(X_1) + H(X_2)}, \quad (3)$$

with random variables X_1, X_2 , the entropy $H(\cdot)$ and the conditional entropy $H(\cdot|\cdot)$. We average the results over the natural image dataset Kodak [15] with 24 images and the SC image dataset SCID [16] with 40 images. To evaluate Y-Cb and Y-Cr dependencies, the chroma planes are upsampled using nearest neighbor.

We can see that SC data shows higher mutual information between channels than the natural images from Kodak in general. The YCbCr color transform removes a lot of cross-channel dependency for both natural and SC data. However, the mutual information between Cb and Cr is still relatively

high for SC images. Consequently, we adapt SCF coding for YCbCr 4:2:0 as follows: First, the adaptive color transform is no longer used since the images are already in the YCbCr color space. Second, we first code the Y channel with the SCF algorithm, i.e., each color contains only the luminance with $c_Y^Y = [c_Y]$. Afterwards, the combined Cb-Cr planes are coded, i.e., $c^{CbCr} = [c_{Cb}, c_{Cr}]^T$, since Cb and Cr still have a lot of interdependencies. Thus, for the Y channel, each pixel ‘color’ contains only one component, for chroma two. Since in 4:2:0 format the chroma components are subsampled by a factor of two both horizontally and vertically, we code only one fourth the number of pixels in the chroma plane. The coder for separate Y and CbCr coding will be called ‘SCF 420’ in the following.

B. Luma-guided Prediction

Chroma prediction based on the luma plane has already been shown to have a positive effect [11,12]. However, for SC in particular, the mutual information between luma and chroma is still relatively high, see Table I. SC has often very clear structures, e.g., horizontal or vertical structures with constant colors, such as single colored text on white background. We assume that the chroma planes are similarly structured as the downsampled luma channel. Accordingly, if a pixel in the luma channel can be perfectly predicted by its top or left neighbor, the same is likely true for the chroma channel. To find correlations between luma and chroma, we first subsample the luma plane to the same resolution as chroma, which is also done in other approaches where luma is used for chroma prediction in 4:2:0 format, e.g., in the cross-component linear model (CCLM) for chroma prediction [12]. Let $Y[m, n]$ of size $M \times N$ with spatial indices (m, n) be the luminance plane of the image in YCbCr 4:2:0 format and $\tilde{Y}[m, n]$ the downsampled version of size $M/2 \times N/2$. We calculate $\tilde{Y}_s[m', n']$ at position (m', n') with a precision scaling parameter s as

$$\tilde{Y}_s[m', n'] = \left\lfloor \frac{1}{s} \left(\sum_{i=0}^1 \sum_{j=0}^1 Y[2m' + i, 2n' + j] + \frac{s}{2} \right) \right\rfloor. \quad (4)$$

Let $C[m', n', c]$ of size $M/2 \times N/2 \times 2$ with spatial indices (m', n') and color channel index $c \in \{Cb, Cr\}$ be the chroma planes. For subsampling the luma plane, we use the precision scaling factor $s = 2$ and enhance the MAP prediction with luma guidance according to

$$\begin{aligned} \text{LMAP}(C[m', n', c]) = & \\ \begin{cases} C[m' - 1, n', c], & \text{if } \tilde{Y}_2[m', n'] = \tilde{Y}_2[m' - 1, n'], \\ C[m', n' - 1, c], & \text{if } \tilde{Y}_2[m', n'] = \tilde{Y}_2[m', n' - 1], \\ \text{MAP}(C[m', n', c]), & \text{otherwise,} \end{cases} \end{aligned} \quad (5)$$

where $\text{MAP}(C[m', n', c])$ is the modified median adaptive prediction as described in [8].

In Table II, we show the effect of LMAP by calculating the mean absolute error (MAE) between prediction, i.e., LMAP or MAP, and true chroma planes. Furthermore, we count how

TABLE II
EFFECT OF PIXEL-WISE PREDICTION METHODS MAP AND LMAP ON CHROMA PLANES.

Data set	MAP		LMAP	
	MAE	Match in %	MAE	Match in %
Kodak [15]	0.708	42.4%	0.711	44.3%
SCID [16]	1.317	81.3%	1.175	82.7%

often the prediction matches the actual chroma values for both Cb and Cr, i.e.,

$$\text{LMAP}(C[m', n', c]) = C[m', n', c] \quad \forall c \in \{Cb, Cr\}. \quad (6)$$

We calculate the ratio of perfect predictions for all positions (m', n') , where one of the first two conditions of (5) is met. We average the results over datasets Kodak and SCID. We can see that the prediction for SC has higher MAE values with both prediction methods than the Kodak natural image dataset, i.e., the prediction is generally worse for SC images. However, LMAP manages to improve the MAE for SC images, while the natural images remain mostly unaffected. Furthermore, with LMAP 82.7% of Cb-Cr color doubles are correctly predicted if the luma pixel could be perfectly predicted by its top or left neighbor. Since we have a high percentage of perfect matches at pixels where LMAP was used, we additionally double counts of chroma values corresponding to the predicted chroma color c^{CbCr} in the merged histogram of Stage 1 and the color palette in Stage 2, whereby increasing their estimated probabilities in the probability distributions.

C. Luma-dependant Chroma Ranges

The number of unique colors in SC images is often relatively small. Consequently, only few out of all possible YCbCr triplets actually occur in the image. Specifically, for each luma value in the subsampled luma image, many Y-Cb or Y-Cr combinations never happen in a specific image. At each position (m', n') , we evaluate the corresponding luminance value. With knowledge of which Y-Cb or Y-Cr combinations does and does not occur in the image, we can exclude chroma color doubles which cannot happen in conjunction with this luminance value from the merged histograms, the color palette or the residual coding histograms. This leads to improved compression efficiency as shown in [9]: Let N_{tot} be the summed counts of all entries in a histogram, e.g., the color palette, the merged histogram or residual histogram. Then, the probability of the i -th entry x_i with count N_i in the histogram is estimated as

$$p(x_i) \approx \frac{N_i}{N_{\text{tot}}}. \quad (7)$$

If impossible entries are removed from the histogram, the new total counts of entries $N_{\text{tot,new}}$ will be the same or smaller than before. As such, for the new estimated probability of the i -th entry, we have

$$p(x_i) \approx \frac{N_i}{N_{\text{tot,new}}} \geq \frac{N_i}{N_{\text{tot}}}. \quad (8)$$

With the higher probability for the i -th entry, less bits are needed to transmit the corresponding symbol using an arithmetic coder.

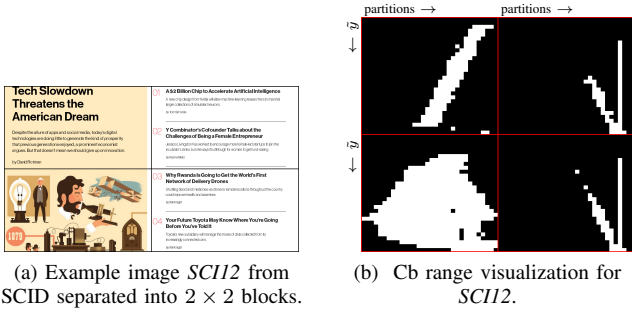


Fig. 3. Example of Cb range visualization for 2×2 blocks, $s = 4$ and $p = 32$ for the corresponding blocks of image *SC12*.

To enable this feature, however, information about occurring Y-Cb and Y-Cr combinations has to be transmitted as side information, which can be very expensive. To reduce the number of bits necessary for this information, we use a downsampled luma image $\tilde{Y}_{s_{CRC}}[m', n']$ with the precision scaling parameter s_{CRC} , i.e., the higher the downsampling factor, the less the maximum luma value \tilde{y}_{max} . Furthermore, we split the possible Cb and Cr ranges into p partitions, where each partition covers a part of the entire Cb or Cr range. The i -th partition for Cb, for example, covers the range $\{r_{min}(i), \dots, r_{max}(i)\}$, where

$$\begin{aligned} r_{min}(i) &= \left\lfloor \frac{i \cdot (c_{Cb, max} + 1)}{p} \right\rfloor, \\ r_{max}(i) &= \left\lfloor \frac{(i+1) \cdot (c_{Cb, max} + 1)}{p} \right\rfloor - 1. \end{aligned} \quad (9)$$

Since the possible Y-Cb or Y-Cr combinations may vary strongly spatially, we also introduce a parameter b , which allows us to split the chroma planes into $b \times b$ blocks of equal size and to calculate the possible Y-Cb and Y-Cr combinations for each block separately. Then, for each block, we check which combinations of quantized luma sample and Cb or Cr, respectively, exist. Figure 3b shows the visualization of the Y-Cb combinations for an example image with 2×2 blocks. For each block, each row in the visualization is a different \tilde{y} value, with $\tilde{y} \in [0, \tilde{y}_{max}]$. Each column depicts a partition of the Cb range. White points indicate that this combination occurs in the block, black points, that they do not occur. In the example, it is visible, that the Y-Cb combinations vary strongly depending on the block, and for many luminance values, only a small part of the possible chroma range does actually occur. To transmit the chroma ranges, we use the original SCF algorithm and transmit both Y-Cb and Y-Cr ranges as small binary images as exemplarily shown in Figure 3b. This method will be called Chroma Range Coding ‘CRC’ in the following. An overview of the entire 4:2:0 coding method is visualized in Figure 4.

IV. EVALUATION

To evaluate SCF coding on 4:2:0 data, we regard five different image datasets with a total of 173 images, namely the data sets SCID [16], SIQAD [17], the ‘mixed’ and ‘textual’ stimuli from [18], which depict webpages, as well as the first

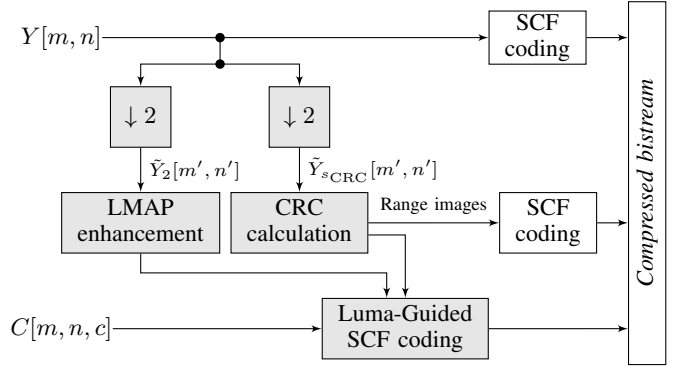


Fig. 4. Overview of the proposed SCF 420 encoding pipeline. Light gray blocks indicate new blocks or modifications.

TABLE III
CRC PARAMETERS OBTAINED BY TRAINING ON SCID DATA SET [16].

Parameter	Values	Selection
Number of blocks b	1, 2, 4	4
Precision scaling parameter s_{CRC}	2, 4, 8, 16, 32, 64, 128	64
Partitions p	4, 8, 16, 32, 64, 128	64

frames from the HEVC common test conditions (CTC) dataset for SC [19]. The datasets are available in RGB 4:4:4 format. We first transform the images to YCbCr according to ITU-R BT.709 [20], and downsample the images from 4:4:4 to 4:2:0 using ffmpeg [21].

First, we chose optimal parameters b , p and s_{CRC} for CRC. To this end, we evaluate the dataset SIQAD with multiple parameter combinations as visualized in Table III. We chose the parameter combination with the best average bitrate as $b = 4$, $p = 64$ and $s_{CRC} = 64$.

Finally, we evaluate our resulting method in Table IV. We compare our method against the lossless VVC, i.e., its reference codec VTM 17.2 [22] with its screen content configuration in lossless mode according to the CTCs [23]. Additionally, we compare against HM-16.21+SCM-8.8, the reference codec for HEVC with its SC coding extension [19]. Our proposed method outperforms both VVC and HEVC for every dataset, with HEVC needing between 3.43% and 10.31% more bitrate than our proposed method. On average, HEVC needs 5.66% more bitrate.

To evaluate the effect of each proposed modification, we carry out an ablation study, wherein we evaluate SCF 420 without CRC and SCF 420 with neither CRC nor LMAP in Table V. The SCF 420 without CRC needs 0.56% more bitrate. Without both LMAP and CRC, SCF 420 is 0.81% worse on average compared to our proposed method.

In Table VI, we evaluate coding times for SCF 420. We run all codecs on the same device and average the encoding and decoding times over all images. It is important to note that SCF 420 is currently not optimized for efficiency or speed. While the encoding time is still slightly slower than HEVC, it is about ten times faster than the current video coding standard VVC. Since SCF coding is a symmetric codec, it needs approximately the same time for encoding and decoding and thus its decoding times are not as fast as HM and VTM. This

TABLE IV
AVERAGE BITRATES IN BPP FOR DIFFERENT LOSSLESS CODING
ALGORITHMS PLUS PERCENTAGES W.R.T. SCF 420

Dataset	VTM 17.2	HM-16.21 SCM-8.8	SCF 420 (proposed)
SIQAD [17]	2.813	2.715	2.536
Percentage	110.04	107.07	100.00
SCID [16]	2.267	2.162	2.077
Percentage	109.13	104.07	100.00
SC-Text [18]	1.307	1.262	1.144
Percentage	114.20	110.31	100.00
SC-Mixed [18]	1.346	1.329	1.285
Percentage	104.69	103.43	100.00
HEVC CTC [19]	2.136	2.063	1.968
Percentage	108.53	104.81	100.00
All	1.728	1.672	1.582
Percentage	109.22	105.66	100.00

TABLE V
ABLATION STUDY: AVERAGE BITRATE IN BPP PLUS PERCENTAGES W.R.T.
THE PROPOSED METHOD.

	SCF 420 (proposed)	SCF 420 w/o CRC	SCF 420 w/o CRC and LMAP
Bitrate in bpp	1.582	1.591	1.595
Percentage	100.00	100.56	100.81

could be alleviated in future work by implementing a dedicated binary range image coding technique with less complexity and applying more sophisticated search algorithms for color-palette coding and context-based coding in comparison to the currently implemented sequential search.

V. SUMMARY

SCF is a state-of-the-art lossless SC coding technique. Until now, it could only compress 4:4:4 content natively and was thus unsuitable for use cases where 4:2:0 compression is necessary. In this paper, we have shown that it can effectively be extended towards 4:2:0 format by separately encoding Y and CbCr planes. We further improve SCF coding for YCbCr 4:2:0 data using a luma-guided enhanced prediction for chroma samples. Additionally, we propose CRC, where we transmit knowledge about occurring luma and chroma combinations as side-information. This allows us to estimate improved color probability models for compression of the chroma planes. In total, our proposed method outperforms HEVC on all evaluated SC data sets with HEVC needing 5.66% bitrate more on average than our proposed method.

REFERENCES

- [1] G. Sullivan, J. Ohm, W. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.
- [2] J. Xu, R. Joshi, and R. Cohen, "Overview of the emerging HEVC screen content coding extension," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 1, pp. 50–62, Jan. 2016.
- [3] B. Bross *et al.*, "Overview of the versatile video coding (VVC) standard and its applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 10, pp. 3736–3764, Oct. 2021.
- [4] T. Nguyen *et al.*, "Overview of the screen content support in VVC: Applications, coding tools, and performance," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 10, pp. 3801–3817, Oct. 2021.

TABLE VI
AVERAGE CODING TIMES IN SECONDS.

	Enc. time	Dec. time
VTM 17.2	153.0	0.1
HM 21.18 SCF-8.8	5.8	0.1
SCF 420 (prop.)	16.8	15.8

- [5] W. Peng *et al.*, "Overview of screen content video coding: Technologies, standards, and beyond," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 6, no. 4, pp. 393–408, Dec. 2016.
- [6] J. Sneyers and P. Wuille, "FLIF: Free lossless image format based on maniac compression," in *Proc. IEEE International Conference on Image Processing (ICIP)*, Sep. 2016, pp. 66–70.
- [7] M. Weinberger, G. Seroussi, and G. Sapiro, "The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS," *IEEE Transactions on Image Processing*, vol. 9, no. 8, pp. 1309–1324, Aug. 2000.
- [8] T. Strutz and P. Möller, "Screen content compression based on enhanced soft context formation," *IEEE Transactions on Multimedia*, vol. 22, no. 5, pp. 1126–1138, May 2020.
- [9] H. Och, S. Uddehal, T. Strutz, and A. Kaup, "Enhanced color palette modeling for lossless screen content compression," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2024, pp. 3670–3674.
- [10] X. Xu *et al.*, "Intra block copy in HEVC screen content coding extensions," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 6, no. 4, pp. 409–419, Dec. 2016.
- [11] J. Huo, D. Wang, H. Yuan, S. Wan, and F. Yang, "Adaptive chroma prediction based on luma difference for H.266/VVC," *IEEE Transactions on Image Processing*, vol. 32, pp. 6318–6331, Nov. 2023.
- [12] W. Li, X. Jiang, T. Song, and D. Han, "Adaptive cross component linear model for chroma intra-prediction in VVC," in *Proc. International Conference on Communications and Broadband Networking (ICCBN)*, Feb. 2022, pp. 52–59.
- [13] H. Och, T. Strutz, and A. Kaup, "Optimization of probability distributions for residual coding of screen content," in *Proc. International Conference on Visual Communications and Image Processing (VCIP)*, Dec. 2021, pp. 1–5.
- [14] P. Möller and T. Strutz, "Improved lossless image compression using adaptive image rotation," in *Proc. European Signal Processing Conference (EUSIPCO)*, Sep. 2019, pp. 1–4.
- [15] R. Franzen. Kodak lossless true color image suite. [accessed 2025-02]. [Online]. Available: <https://r0k.us/graphics/kodak/>
- [16] Z. Ni, L. Ma, H. Zeng, Y. Fu, L. Xing, and K. Ma, "SCID: A database for screen content images quality assessment," in *Proc. International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, Nov. 2017, pp. 774–779.
- [17] H. Yang, Y. Fang, and W. Lin, "Perceptual quality assessment of screen content images," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 4408–4421, Nov. 2015.
- [18] C. Shen and Q. Zhao, "Webpage saliency," in *Proc. European Conference on Computer Vision (ECCV)*. Springer International Publishing, Sep. 2014, pp. 33–46.
- [19] H. Yu, R. Cohen, K. Rapaka, and J. Xu, "Common test conditions for screen content coding," Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JCTVC-U1015, Jun. 2015.
- [20] "ITU-R BT.709-6: Parameter values for the HDTV standards for production and international programme exchange," International Telecommunication Union, Technical Report, Jun. 2015.
- [21] (2024) Fast Forwards MPEG (FFmpeg). [accessed 2024-02]. [Online]. Available: <http://ffmpeg.org>
- [22] Joint Video Experts Team (JVET). VVC test model reference software (VTM). [accessed 2024-02]. [Online]. Available: https://vcgit.hhi.fraunhofer.de/jvet/VVCSsoftware_VTM
- [23] T. Nguyen, T. Ma, and A. Nalci, "JVET common test conditions and software reference configurations for lossless, near lossless, and mixed lossy/lossless coding," Joint Video Exploration Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, AHG Report, JVET-Q2014, Jan. 2020.