Tunneling High-Resolution Color Content through 4:2:0 HEVC and AVC Video Coding Systems

Yongjun Wu, Sandeep Kanumuri, Yifu Zhang, Shyam Sadhwani,
Gary J. Sullivan, and Henrique S. Malvar

Microsoft Corporation
One Microsoft Way
Redmond, WA 98052, USA

Abstract: We present a method to convey high-resolution color (4:4:4) video content through a video coding system designed for chroma-subsampled (4:2:0) operation. The method operates by packing the samples of a 4:4:4 frame into two frames that are then encoded as if they were ordinary 4:2:0 content. After being received and decoded, the packing process is reversed to recover a 4:4:4 video frame. As 4:2:0 is the most widely supported digital color format, the described scheme provides an effective way of transporting 4:4:4 content through existing mass-market encoders and decoders, for applications such as coding of screen content. The described packing arrangement is designed such that the spatial correspondence and motion vector displacement relationships between the nominally-luma and nominally-chroma components are preserved. The use of this scheme can be indicated by a metadata tag such as the frame packing arrangement supplemental enhancement information (SEI) message defined in the HEVC and AVC (Rec. ITU-T H.264 | ISO/IEC 14496-10) video coding standards. In this context the scheme would operate in a similar manner as is commonly used for packing the two views of stereoscopic 3D video for compatible encoding. The technique can also be extended to transport 4:2:2 video through 4:2:0 systems or 4:4:4 video through 4:2:2 systems.

1. Introduction

Most video codecs that are commercially available today support only the 4:2:0 chroma format [1], in which the chroma resolution is half that of the luma resolution both vertically and horizontally, as contrasted with using a 4:4:4 format, in which the chroma information is represented at the same resolution used for the luma [1]. The YC_BC_R (a.k.a. YUV) 4:2:0 format is good enough for "mainstream" content (i.e. most camera-view, animation, and gaming content), for which users do not ordinarily see a perceptible difference between the two formats. However, there are a variety of existing and emerging applications, such as cloud computing, cloud-mobile computing, remote desktop, virtual desktop infrastructure, thin client, and wireless displays, which operate with "screen content" [2] that includes hard-edged text and graphics. For such applications, the difference between the 4:4:4 and 4:2:0 color formats can be more visually perceptible, as shown in Fig. 1.

Codecs designed specifically for screen content encode color in full 4:4:4 resolution. One example is Microsoft RemoteFX [3], for which there are no visually-perceptible

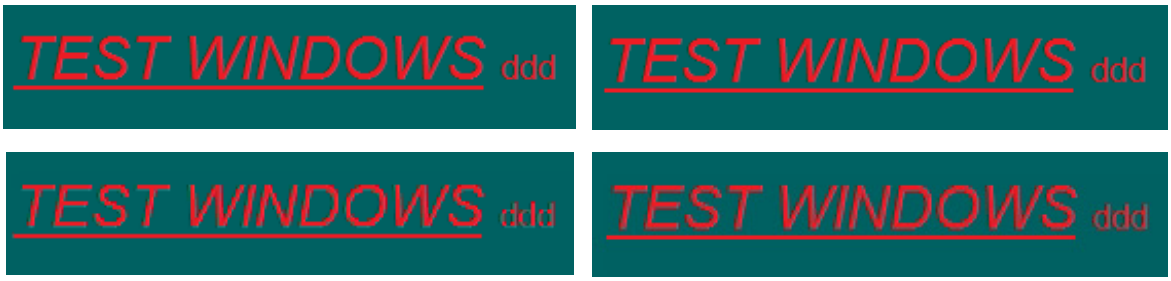

Figure 1. Top left: original screen capture in RGB color space at 288×60 resolution. Top right: same screen with color converted from 8-bit-per-sample RGB to 8-bit YUV 4:4:4 and back to 8-bit RGB. Bottom left: same screen with color converted from RGB to YUV 4:4:4 and then to YUV 4:2:0 with no chroma anti-alias prefiltering, then upsampled from YUV 4:2:0 to YUV 4:4:4 via pixel replication, and finally converted from YUV 4:4:4 back to RGB. Bottom right: same steps as in the bottom left, but with 2×2 pixel averaging when downsampling to YUV 4:2:0. We see that conversion from RGB to YUV 4:4:4 at full resolution has no perceptible distortion, whereas conversion to YUV 4:2:0 can lead to aliasing and blurring artifacts, depending on the filtering steps used for downsampling and upsampling.

artifacts for graphic content with sharp color transitions, including color text (such as the clip in Fig. 1) and text with fine color fringes generated by effects such as ClearType font rendering [4]. RemoteFX is fast and performs well for remote desktop applications, thanks to its combined use of central and graphics processors (CPU and GPU).

However, for many video applications, such as mobile entertainment and video conferencing, a specialized hardware module supporting a general-purpose standard codec is available in the system, such as the Baseline or High profile of the popular H.264/MPEG-4 AVC standard [5][6]. The Main profile of the emerging High Efficiency Video Coding (HEVC) standard will soon have a similar broad deployment status [7][8]. In such contexts, faster processing and significantly lower power consumption would be achieved in remote desktop and similar applications if screen content can be processed in dedicated chips. However, such hardware modules typically support only the 4:2:0 format profiles of the standard, and thus cannot be directly used for 4:4:4 applications.

We present an approach for leveraging codecs designed for YUV 4:2:0 content to compress and represent 4:4:4 content with good fidelity, through the use of content splitting and frame packing. This method has some similarity to the frame packing of stereo (3D) content into 2D images, and builds on that framework by extending the semantics of the frame packing arrangement (FPA) supplemental enhancement information (SEI) messageas specified in [6]. Unlike frame packing of stereo content, for which there is a “left” and “right” view, we introduce frame packing of 4:4:4 content via a “main view” and an “auxiliary view”, both represented in 4:2:0 format. This allows for full compatibility with conventional 4:2:0 encoding, as decoding the main view leads to a 4:2:0 representation of the original video. When full 4:4:4 resolution is desired, data from the main view can be combined with data of the auxiliary view to form a full resolution 4:4:4 color format representation. This work expands on a scheme originally described in contributions to the JCT-VC committee for development of the HEVC standard (in which we also proposed having the same extension to the AVC standard as well) [9][10].

2. Packing a YUV 4:4:4 frame into main and auxiliary views

A frame in YUV (i.e., YC_BC_R, YCoCg, GBR, etc.) 4:4:4 format [1] can be represented as indicated in the top part of Fig. 2, where Y_{444}, U_{444}, and V_{444} are the Y, U, and V planes comprising the YUV 4:4:4 frame.

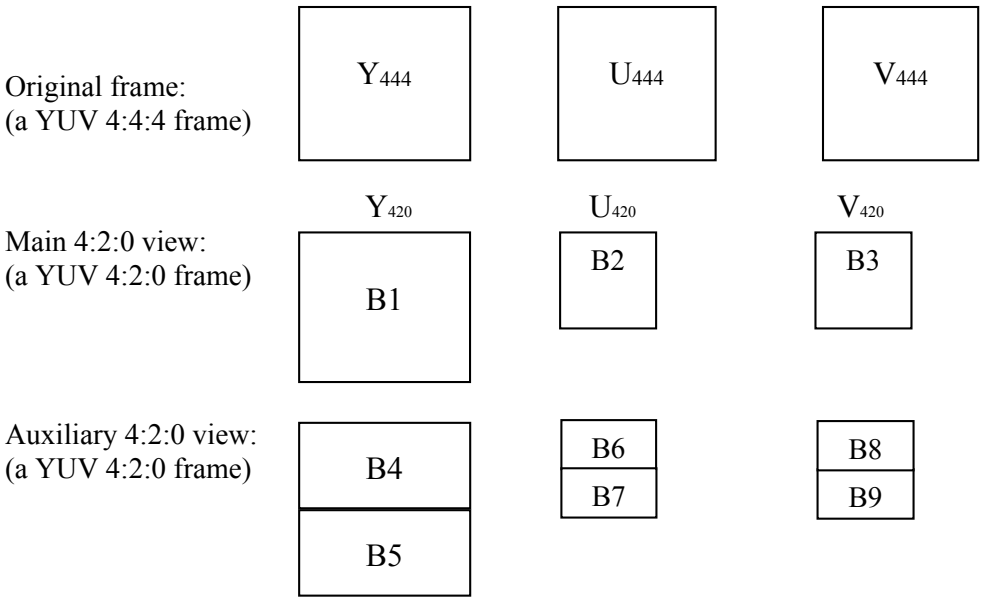

Figure 2. Top: Representaion of an original frame in YUV 4:4:4 format. Bottom: Decomposition of the frame into two YUV 4:2:0 views.

Let the resolution of these planes be represented by width W and height H. The YUV 4:4:4 frame represented above can be packed into two YUV 4:2:0 frames (as main and auxiliary view frames) as shown in the bottom part of Fig. 2. The areas marked as B1 to B9 make up the Y, U and V planes of the two YUV 4:2:0 frames representing the main and auxiliary views. These areas can be related to Y_{444}, U_{444}, and V_{444} as follows:

Area B1: $Y_{420}^{main}(x,y) = Y_{444}(x,y)$, where the range of (x,y) is $[0, W-1] \times [0, H-1]$.

Area B2: $U_{420}^{main}(x,y) = \widetilde{U}_{444}(2x,2y)$, with (x,y) in $\left[0, \frac{W}{2}-1\right] \times [0, \frac{H}{2}-1]$.

Area B3: $V_{420}^{main}(x,y) = \widetilde{V}_{444}(2x,2y)$, with (x,y) in $\left[0, \frac{W}{2}-1\right] \times [0, \frac{H}{2}-1]$.

Area B4: $Y_{420}^{aux}(x,y) = U_{444}(x,2y+1)$, with (x,y) in $[0, W-1] \times [0, \frac{H}{2}-1]$.

Area B5: $Y_{420}^{aux}\left(x, \frac{H}{2}+\ y\right) = V_{444}(x,2y+1)$, with (x,y) in $[0, W-1] \times [0, \frac{H}{2}-1]$.

Area B6: $U_{420}^{aux}(x,y) = U_{444}(2x+1,4y)$, with (x,y) in $\left[0, \frac{W}{2}-1\right] \times [0, \frac{H}{4}-1]$.

Area B7: $U_{420}^{aux}\left(x, \frac{H}{4}+y\right) = V_{444}(2x+1,4y)$, with (x,y) in $\left[0, \frac{W}{2}-1\right] \times [0, \frac{H}{4}-1]$.

Area B8: $V_{420}^{aux}(x,y) = U_{444}(2x+1,4y+2)$, with (x,y) in $\left[0, \frac{W}{2}-1\right] \times [0, \frac{H}{4}-1]$.

Area B9: $V_{420}^{aux}\left(x, \frac{H}{4}+y\right) = V_{444}(2x+1,4y+2)$, with (x,y) in $\left[0, \frac{W}{2}-1\right] \times [0, \frac{H}{4}-1]$.

In the above equations, $\widetilde{U}_{444}(2x, 2y)$ and $\widetilde{V}_{444}(2x, 2y)$ are either the same as or represent anti-alias filtered values corresponding to $U_{444}(2x, 2y)$ and $V_{444}(2x, 2y)$, respectively, where the range of (x, y) is $[0, W/2 - 1] \times [0, H/2 - 1]$. This choice is explained in more detail in section 4. The packing method is designed such that the main view is the YUV 4:2:0 equivalent of the original YUV 4:4:4 frame. Systems can just display the main view if YUV 4:4:4 is either not supported or is considered not necessary for the decoder.

The auxiliary view fits the content model of a YUV 4:2:0 frame and is well suited for compression in this manner, in terms of the spatial position consistency across its Y, U and V components (which is useful for such purposes as spatial block size segmentation and joint coding of coded block pattern signaling) and in terms of the motion displacement correspondence across its Y, U and V components (e.g., a vertical or horizontal displacement of 2 samples in Y corresponds to a displacement of 1 sample in U and V, as in ordinary 4:2:0 video).

Here we have described the packing of 4:4:4 content into 4:2:0 frames. It is easily shown that with small adjustments, the same concept can also be used to pack 4:4:4 content into 4:2:2 frames (i.e. frames with half-horizontal resolution and full vertical resolution for the chroma) or to pack 4:2:2 content into 4:2:0 frames.

3. Extension to frame packing arrangement SEI message

To signal the frame packing of YUV 4:4:4 content, we have proposed [9][10] to extend the frame packing arrangement (FPA) SEI message found in the AVC [5][6] and HEVC [7][8] specifications. In particular, the element "`content_interpretation_type`" could be interpreted as shown in Table 1 [9][10], in which the specification of new values 3 and 4 has been added. From a standardization perspective, the following usage extension of the SEI message syntax is proposed to signal the use of main and auxiliary views:

1. When `content_interpretation_type` is equal to 3 to 4, the following syntax elements would be required to be set to 0:
    - `quincunx_sampling_flag`
    - `spatial_flipping_flag`
    - `frame0_grid_position_x`
    - `frame0_grid_position_y`
    - `frame1_grid_position_x`
    - `frame1_grid_position_y`
2. When `content_interpretation_type` is equal to 3, the following syntax elements should be required to be set as follows, since these values represent the correct location type for chroma in the main view in this case:
    - `chroma_loc_info_present_flag` would be equal to 1,
    - `chroma_sample_loc_type_top_field` and `chroma_sample_loc_type_bottom_field` would be equal to 2.
3. Any of several types of frame packing arrangements can be used to convey the main and auxiliary views – for example, any of the following:
    - Side-by-side packing (`frame_packing_arrangement_type` = 3)
    - Top-bottom packing (`frame_packing_arrangement_type` = 4)
    - Temporal interleaving (`frame_packing_arrangement_type` = 5)

Value	Interpretation
0	Unspecified relationship between the frame-packed constituent frames.
1	Two frames are a stereo view scene; 0 = left, 1 = right.
2	Two frames are a stereo view scene; 1 = left, 0 = right.
3 (new)	Two frames form main and auxiliary views (4:2:0 frames) representing a 4:4:4 frame; 0 = main, 1 = auxiliary. Chroma samples of frame 0 are unfiltered samples of the 4:4:4 frame (without anti-alias filtering).
4 (new)	Two frames form main and auxiliary views (4:2:0 frames) representing a 4:4:4 frame; 0 = main, 1 = auxiliary. Chroma samples of frame 0 are samples of the 4:4:4 frame that were anti-alias filtered prior to frame packing.

Table 1. Definition of `content_interpretation_type`. Values 0–2 are defined in the existing AVC and HEVC specifications; values 3 and 4 are extensions proposed to signal the frame packing of YUV 4:4:4 content.

The use of `content_interpretation_type` in the frame packing arrangement (FPA) SEI message with a value equaling 3 or 4 would inform the decoder that the decoded pictures contain main and auxiliary views of a 4:4:4 frame as the constituent frames of the frame packing arrangement. This information can be used to process the main and auxiliary views appropriately for display or other purposes.

When the system at the decoding end desires the video in 4:4:4 format and is capable of reconstructing the 4:4:4 frames from the main and auxiliary views, it should do so and the output format should be 4:4:4. Otherwise, only the main view should be given as output and the output format will then be 4:2:0.

4. Pre-processing and post-processing considerations

With the proposed extensions to the frame packing arrangement SEI message described in section 3, we also enable the flexibility of applying pre-processing and post-processing on 4:4:4 chroma samples. The distinction between the proposed values 3 and 4 is an indication of whether pre-processing has been applied by the decoder.

4.1. No pre-processing and post-processing

When `content_interpretation_type` is set to 3, the indication would be that none of the chroma samples underwent an anti-alias filtering operation during the process of frame packing i.e. $\widetilde{U}_{444}(2x, 2y) = U_{444}(2x, 2y)$ and $\widetilde{V}_{444}(2x, 2y) = V_{444}(2x, 2y)$. In such a case, the chroma samples comprising the main view are a result of a direct sub-sampling of the chroma planes representing the 4:4:4 frame. However, as shown in the Fig. 1, direct sub-sampling without filtering can create aliasing artifacts for certain types of screen content when only the main view is used to generate a 4:2:0 output.

4.2. Anti-alias filtering

In order to reduce the aliasing artifacts and improve the visual quality for the case where only the main view is used, `content_interpretation_type` can be set to 4 and the main view can be generated using filtered versions of the 4:4:4 chroma planes. In such a

case, the filter choice should be made based on the chroma sample grid alignment with luma sample grid (inferred from `chroma_sample_loc_type_top_field` and `chroma_sample_loc_type_bottom_field`). For simplicity, in the case when the chroma sample grid aligns with the luma sample grid for each direction (horizontal and vertical), it is suggested that that the 3-tap filter [1 2 1] / 4 be used in that direction. If the chroma sample grid positions are centered between the luma sample positions for a particular direction (horizontal/vertical), then it is suggested that the 2-tap filter [1 1] / 2 be used in that direction. Another possible filter choice for the latter case is [1 3 3 1] / 8.

For example, when we consider the case where the chroma sample grid is not aligned with the luma sample grid, in both the horizontal and vertical directions (which corresponds to setting the values of both `chroma_sample_loc_type_top_field` and `chroma_sample_loc_type_bottom_field` equal to 1), the 2-tap filter [1 1] / 2 is applied in both directions, so that $\widetilde{U}_{444}(2x, 2y)$ and $\widetilde{V}_{444}(2x, 2y)$ are obtained by:

$$\widetilde{U}_{444}(2x, 2y) = \frac{[U_{444}(2x, 2y) + U_{444}(2x+1, 2y) + U_{444}(2x, 2y+1) + U_{444}(2x+1, 2y+1) + 2]}{4}$$

$$\widetilde{V}_{444}(2x, 2y) = \frac{[V_{444}(2x, 2y) + V_{444}(2x+1, 2y) + V_{444}(2x, 2y+1) + V_{444}(2x+1, 2y+1) + 2]}{4}$$

When pre-processing is used (`content_interpretation_type` set to 4), the main view does not contain samples $U_{444}(2x, 2y)$ and $V_{444}(2x, 2y)$ but instead contains their filtered counterparts $\widetilde{U}_{444}(2x, 2y)$ and $\widetilde{V}_{444}(2x, 2y)$. The auxiliary view contains the other chroma samples (without any pre-filtering).

If the decoding system decides to output a 4:4:4 frame, a post-processing step should be applied to estimate the samples $U_{444}(2x, 2y)$ and $V_{444}(2x, 2y)$ as $U'_{444}(2x, 2y)$ and $V'_{444}(2x, 2y)$ from the decoded packed frame. For example, a simple suggested formula for deriving $U'_{444}(2x, 2y)$ and $V'_{444}(2x, 2y)$ from decoded representations of the encoded input data (with lossy coding denoted by a hat symbol) would be:

$$\begin{aligned} U'_{444}(2x, 2y) = {} & (1 + \alpha + \beta + \gamma) \cdot \widehat{\widetilde{U}}_{444}(2x, 2y) - \alpha \cdot \widehat{U}_{444}(2x+1, 2y) \\ & - \beta \cdot \widehat{U}_{444}(2x, 2y+1) - \gamma \cdot \widehat{U}_{444}(2x+1, 2y+1) \\ V'_{444}(2x, 2y) = {} & (1 + \alpha + \beta + \gamma) \cdot \widehat{\widetilde{V}}_{444}(2x, 2y) - \alpha \cdot \widehat{V}_{444}(2x+1, 2y) \\ & - \beta \cdot \widehat{V}_{444}(2x, 2y+1) - \gamma \cdot \widehat{V}_{444}(2x+1, 2y+1)] \end{aligned}$$

In the proposed form, setting the value of `content_interpretation_type` equal to 4 and setting the values of both `chroma_sample_loc_type_top_field` and `chroma_sample_loc_type_bottom_field` equal to 1, with the suggested anti-alias filter [1 1] / 2, then the values $\alpha = \beta = \gamma = 1$ would perfectly reconstruct the input values in the absence of quantization error and rounding error. When considering quantization error, to reduce artifacts, smaller values of these parameters should be used (e.g., $\alpha = \beta = \gamma = 0.5$). In general α, β and γ should be in the range from 0.0 to 1.0, and should be smaller for lower-fidelity coding (e.g. coding with larger quantization step sizes). The values of α, β and γ can, e.g., be designed for conditional optimality using cross-correlation analysis.

4.3. Frequency band separation for the auxiliary frame

In the pre- and post- processing methods in section 4.1 and 4.2, pixel values of the U_{444} and V_{444} frames are placed directly into (and are directly unpacked from) the auxiliary frames. We thus refer to these schemes as "direct" packing approaches. Alternatively, we can consider the auxiliary frame samples as an enhancement layer signal to be combined with the main frame (or base layer frame) data. The main and auxiliary frame data can formed using low-pass and high-pass band separation filtering, instead of direct sample packing. With this variation, the primary signal energy can be concentrated into the main frame, and arbitrarily low bit rates can be allocated to the supplemental auxiliary frame data that forms the enhancement signal.

Instead of encoding auxiliary frame pixels directly, a two-dimensional, three-band wavelet decomposition can first be applied to U_{444} and V_{444} before the actual encoding process. A typical four-band wavelet decomposition breaks the frame into "LL", "LH", "HL" and "HH" subbands ("LL" = low-pass in both vertical and horizontal directions, "LH" = low-pass vertical, high-pass horizontal, and so forth). In our wavelet packing scheme, though, the "HL" and "HH" bands are not created; instead, the vertical high-pass signal is kept at full horizontal resolution, i.e., B2 and B3 are the "LL" bands of U_{444} and V_{444} respectively, B4 and B5 are vertical high-pass signals, i.e. a vertical "H" band of U_{444} and V_{444}, respectively, B6 and B8 consist of even-numbered rows of the "LH" band of U_{444}, and B7 and B9 consist of odd-numbered rows the "LH" band of V_{444}. That way, the decoder would apply the corresponding inverse wavelet operations after decoding the main and auxiliary frames to obtain U_{444} and V_{444} pixels. Moreover, an additional vertical band separation can be performed, such that B6 and B8 are an "LHL" and "LHH" band of U_{444}, and B7 and B9 are an "LHL" and "LHH" band of V_{444}.

When the auxiliary frames are transmitted at lower bit rates (lower quality relative to the main frame), the chroma information from the main frame (U_{420}^{main} and V_{420}^{main}) sets the minimum level of quality for the U_{444} and V_{444} reconstruction, and any information from the auxiliary frame is used to improve beyond that minimum quality level. In the case of the "direct" frame packing method, wherein pixels from the auxiliary frame are directly unpacked into U_{444} and V_{444} frames, such an approach would cause the chroma pixels obtained from the main frame (3 out of 4) to have a lower quality compared to the chroma pixels obtained from the main frame. However, the band-separation frame packing approach incurs a larger rounding error in the pre-processing steps than the direct frame packing approach because of the additional filtering operations involved (in the absence of bit-depth expansion).

The rounding error could be reduced or eliminated via lifting implementations of the sub-band filters, possibly in combination with clipping to avoid dynamic range expansion. For example, we may use a lifting-based Haar wavelet decomposition to construct vertical low-pass and clipped high-pass signals $L(x, y)$ and $H(x, y)$ for an input video signal $S(x, y)$ with a dynamic range of 0 to $2^B - 1$, using a temporary variable *t*:

$$t = S(2x, y) - S(2x + 1, y)$$

$$L(x, y) = S(2x + 1, y) + (t \gg 1)$$

$$H(x, y) = Clip_{[0,\, 2^B)}[t + 2^{B-1}]$$

where $(t \gg 1)$ denotes an arithmetic right shift of t by one bit position in two's complement arithmetic, and the function $Clip_{[m,\,n)}[a]$ evaluates to the argument a when $m \leq a < n$, evaluates to m when $a < m$, and evaluates to $n - 1$ when $a \geq n$.

This operation is fully reversible except when the clipping affects the signal, and the low-pass signal has the same dynamic range as the input signal. Expansion of the dynamic range of the high-pass signal is prevented by the clipping (which is applied after constructing the low-pass signal). Although this clipping can introduce distortion, the clipped high-pass signal would still provide a significant enhancement of the low-pass signal, and clipping distortion may rarely occur in practice. Thus, the benefit of eliminating the rounding error may outweigh the detriment of introducing the clipping error. The inverse operations to recover approximations $S'(2x, y)$ and $S'(2x + 1, y)$, starting with decoded approximations $L'(x, y)$ and $H'(x, y)$ which each have a dynamic range of 0 to $2^B - 1$, are:

$$t' = H'(x, y) - 2^{B-1}$$

$$S'(2x + 1, y) = Clip_{[0,\,2^B)}[L'(x, y) - (t' \gg 1)]$$

$$S'(2x, y) = Clip_{[0,\,2^B)}[(t' \gg 1) + S'(2x + 1, y)]$$

If $L'(x, y)$ is equal to $L(x, y)$ and $H'(x, y)$is equal to $H(x, y)$ and $H(x, y)$ does not have a clipped value – i.e. $H(x, y)$ is equal to $t + 2^{B-1}$, then $S'(2x, y)$will be equal to $S(2x, y)$ and $S'(2x + 1, y)$ will be equal to $S(2x + 1, y)$. The above equations show the horizontal processing steps, which would suffice for conversion from 4:4:4 to 4:2:2. For conversion to 4:2:0, the same processing would also be applied vertically, in a cascaded fashion. If the encoder performs the horizontal conversion stage first, the decoder should perform the vertical inverse conversion stage first (to achieve lossless inverse conversion).

5. Experiments

In Fig. 1, the difference in quality between the different variants can be easily seen. The bottom left image has the worst quality, with the bottom right image having slightly better quality than the bottom left – both noticeably worse than the top right image. By using the frame packing scheme, we can achieve quality similar to the top right image in Fig. 1. Without the use of frame packing, chroma artifacts are observed, similar to those in the bottom row images in Fig. 1 (depending on the downsampling filter used when converting from YUV 4:4:4 to YUV 4:2:0).

We first tested an end-to-end system for packing a 4:4:4 frame into two 4:2:0 frames, based on Microsoft's implementation of an AVC encoder and decoder with a simple "IPPP" (forward-predictive) coding structure for an example screen content video test sequence. We also conducted some similar tests using the HM 9.0 reference software HEVC encoder [11]. Each encoder starts with a 4:4:4 input frame, constructs a 4:2:0 frame with twice the height of the 4:4:4 frame, places the main view in the top half and the auxiliary view in the bottom half of the 4:2:0 frame, and encodes the 4:2:0 frame. This corresponds to the use of the top-bottom variation of the FPA SEI message (`frame_packing_arrangement_type` equal to 4) [9][10]. The decoder decodes the 4:2:0 frame, extracts the main and auxiliary views, and reassembles the 4:4:4 frame for output.

We tested both the "direct" frame packing approach (using $\alpha = \beta = \gamma = 1$ to simplify the initial testing) and one variation of band-separation frame packing. The tested band separation approach used a Haar wavelet (i.e., $[1\ 1]/2$ and $[1\ -1]/2$ filtering with rounding). Figs. 3 and 4 show comparisons between these approaches at different bit rates for the auxiliary frame. Each frame is divided into two slices each, for the main and the auxiliary frames. In each case, the band-separation approach performs well at low bit rates for the auxiliary frame, but suffers at high bit rates due to rounding error, while direct frame packing works better at high bit rates, as it introduces no rounding error.

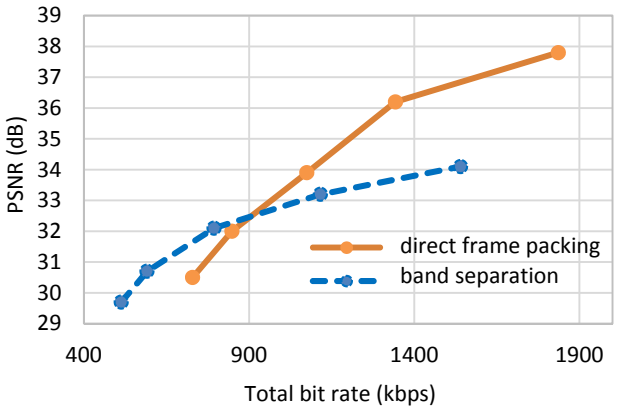

Figure 3. Rate-distortion performance comparison between the direct frame packing and band-separation approaches for a fixed main frame bit rate using a Microsoft AVC encoder with a screen content sequence of resolution 1920×1200 and length 57 frames, at 30 fps. Auxiliary frame QP varies from −12 to +4 relative to main frame QP, which in this case is set to 39. The bit rate for the main view is 445 kbps, with a PSNR of 31.3 dB.

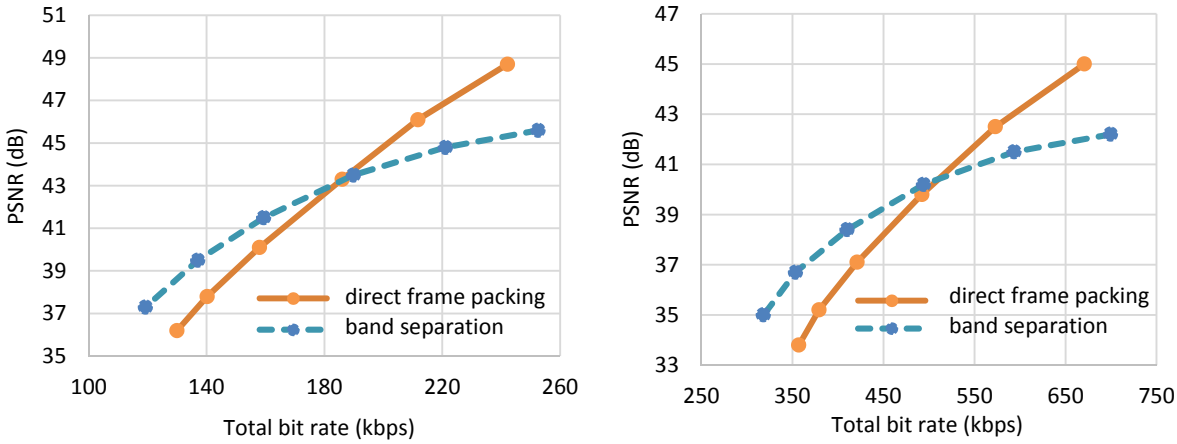

Figure 4. Rate-distortion performance comparison between the direct frame packing and band-separation approaches for a fixed main frame bit rate using the HEVC test model HM 9.0 encoder [11] with two screen content sequences of resolution 1024×768 and length 10 frames, at 30 fps. The auxiliary frame QP varies from −12 to +4 relative to the main frame QP, which in this case is set to 26. The left sequence bit rate for the main view is 45 kbps, with a PSNR of 46.2 dB. The right sequence bit rate for the main view is 127 kbps, with a PSNR of 42.9 dB.

6. Conclusion and future work

We presented frame-packing methods that enable transmission of video or image content with full 4:4:4 chroma resolution through encoding systems designed for 4:2:0 chroma resolution, thus preserving compatibility with existing decoding processes. As 4:2:0 is the most widely supported chroma format in practice, our system provides the substantial benefit of enabling widespread near-term deployment of 4:4:4 high color resolution capability. We are currently exploring other options for pre-processing and post-processing algorithms (including the use of lifting and clipping operations in particular), bit rate allocation, and QP adaptation between the main and auxiliary views. Further work would also be desirable to compare the compression performance of the frame packing methods to that of a more conventional 4:4:4 coding approach such as the 4:4:4 Predictive Profile of AVC. Such a comparison would be helpful to determine which of the approaches is appropriate for an application. However, we are confident that the proposed scheme could often provide the ability to achieve 4:4:4 quality in situations where it would otherwise be necessary to settle for 4:2:0.

References

[1] K. R. Rao and J. J. Hwang, *Techniques* and *Standards for Image, Video, and Audio Coding*. New Jersey: Prentice-Hall, 1996, Chapter 2.

[2] T. Lin, P. Zhang, S. Wang, K. Zhou, and X. Chen, "Syntax and semantics of Dual-coder Mixed Chroma-sampling-rate (DMC) coding for 4:4:4 screen content", document JCTVC-J0233, 10th JCT-VC Meeting: Stockholm, Sweden, July 2012.

[3] Microsoft Corporation, "Microsoft RemoteFX," Available at http://technet.microsoft.com/en-us/library/ff817578(WS.10).aspx, Feb. 2011.

[4] Microsoft Corporation, "ClearType information," Available at http://www.microsoft.com/typography/cleartypeinfo.mspx, Jan. 2010.

[5] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC Video Coding Standard", *IEEE Trans. on Circuits and Systems for Video Tech.*, Vol. 13, No. 7, pp. 560–576, July 2003.

[6] ITU-T and ISO/IEC, *Advanced Video Coding for Generic Audiovisual Services*, Rec. ITU-T H.264 | ISO/IEC 14496-10, Jan. 2012.

[7] 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 and Systems for Video Technology*, Dec. 2012.

[8] B. Bross, W.-J. Han, J.-R. Ohm, G. J. Sullivan, and T. Wiegand "High Efficiency Video Coding (HEVC) text specification draft 9 (SoDIS)", document JCTVC-K1003, 11th JCT-VC meeting, Shanghai, Oct. 2012.

[9] Y. Wu, S. Kanumuri, S. Sadhwani, L. Zhu, S. Sankuratri, G. J. Sullivan, and B. A. Kumar, "Frame packing arrangement SEI for 4:4:4 content in 4:2:0 bitstreams", document JCTVC-K0240, 11th JCT-VC meeting, Shanghai, Oct. 2012.

[10] Y. Zhang, Y. Wu, S. Kanumuri, S. Sadhwani, G. J. Sullivan, and H. S. Malvar, "Updated proposal for frame packing arrangement SEI for 4:4:4 content in 4:2:0 bitstreams", document JCTVC-L0316, 12th JCT-VC meeting, Geneva, Jan. 2013.

[11] "HEVC software repository", https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware.