# UltraHD, HEVC and Higher Fidelity Video
## Why It's Not Just Pixel Density Anymore

Indra Laksono, Cindy Guo
Vixs Systems

*Abstract*

We present the advantages of 10-bit video over the traditional 8-bit video compression. Results on both objective and subjective video quality are discussed along with some highlights of implementation issues for anyone looking at a IC implementation of a 10-bit HEVC decoder. 10-bit video compression can achieve higher coding efficiency while delivering visible quality improvement.

## INTRODUCTION

The deployment of digital video is facilitated by the development of modern video coding standards such as MPEG-2, H.264/MPEG-4 Advanced Video Coding (AVC), and mostly recently, High Efficiency Video Coding (HEVC) otherwise known as H.265. Each generation of adopted video coding standards has offered a target bit rate savings of at least 50%. This enables broadcast industry to deliver video distribution of resolutions from 720x480 or 720x576 (SD), to the current standard of 1920x1080 (HD), and now to 3840x2160 (UltraHD).

As pixel density increases from SD to HD/UltraHD, the basic unit of video is still YUV 4:2:0, with chroma channel subsampled, and each colour components are 8 bits per sample. This so-called "True Colour" system can support more than 16 million colours. While some study suggests that human vision system can only discriminate up to 10 million colours [1], banding artifacts (also known as contouring) can be visible in areas with low chrominance variation as shown in Fig. 1 [2]. The prominence of banding artifacts in distributed video today suggests that the dynamic range of colour representation chosen today in distributed video still has gaps that are discernible to the human eye. Furthermore, with increase of display sizes and with display's wider colour gamut and higher dynamic range, visual artifacts of 8-bit video are exposed more easily.

The limitations of 8-bit video were recognized by the International Telecommunication Union (ITU). In August 2012, ITU released its recommendation for UltraHD TV, commonly known as ITU-R BT.2020. This recommendation is not just a change in the colour space conversion matrix coefficients; it specifies that the bit depth of each colour component will be 10 bits or 12 bits. Taking their cue from ITU, the Joint Collaborative Team on Video Coding (JVT-VC) added a consumer oriented 10-bit profile, named as Main10, in October 2012 [3]. This is notable as this is the first time a major video codec standard, targeted for wide consumer adoption has formally allowed a bit depth higher than 8 bits in the first ratified release.

This paper will discuss the various aspects of 10-bit digital video implementations vs. 8-bit, studying visual differences, bandwidth

Fig. 1: A screen shot from the introduction sequence of "House of Cards". Only luma channel is shown here.

requirements, and some IC implementation issues of supporting 10 bits. Moving from 8-bit colour representation to 10 bits will quadruple the dynamic range of each colour component, and provide a 64 times increase in total dynamic range of different colours. This shift can be accomplished with little incremental cost while delivering a visible tangible improvement to the end user. The resulting implementation will move us from 16 million colours to over 1000 million.

The traditional use of 8 bits for colour representation provides us a 24-bit colour space commonly known as True Colour. Due to the limitation of rounding to the component bit depth, mapping between RGB and YUV colour spaces are not generally reversible. For example, given all 24-bit RGB triplets, only 24% of them can be exactly recovered from converting to 8-bit YUV and back to 8-bit RGB using Rec.709. Because of clipping, most RGB triplets are off by ±1. However, increasing bit depth can improve the density of representation of colours and brightness throughout the entire range of colours. When converting between YUV 10-bit and RGB 8-bit, the recovery rate is 100% because of the extra precision. Hence there are always measurable advantages of higher bit depth in video fidelity.

Colour representations of 30 bits and higher are collectively known as Deep Colour in TV parlance and is available widely in 1080p TVs sold today.

## HIGHER FIDELITY VIDEO

Moving from 8-bit video compression to 10-bit poses several challenges and the primary concern is on coding efficiency. Coding efficiency is measured by the balance between video quality and bit rate. As 10 bits per pixel (bpp) is 25% more than 8 bpp, to get the same subjective or objective video quality, one might question whether delivering 10 bits video would require more data bandwidth. Another concern is on the implementation. For software encoders and decoders, 8 bpp fits nicely into bytes; whereas 10 bpp requires either data packing or doubling memory requirements. On the ASIC side, there are implementation issues on memory bandwidth usage.

In this section, we examine various technical challenges of 10-bit video processing.

### Coding Efficiency

We choose 4 test sequences, which are summarized in Table 1, to test coding efficiency. All sequences are progressive with YUV 4:2:0 colour format and 10 bpp. 8-bit YUVs are generated by rounding up the last two least significant bits (LSB) of the 10-bit sequences. All test materials are coded with HM12.0 reference encoder [4] using the random access configuration but with only 4 B frames (instead of 8) per GOP. Each test sequence is encoded at 4 different bit rates, using quantization parameters (QP) 20, 26, 32 and 38.

Coding efficiency is evaluated subjectively with rate-distortion curves and peak signal-to-noise ratio (PSNR) is chosen as the distortion measure. Fig. 2 shows the rate-distortion curves for the 4 streams, in which Y-PSNR is plotted as a function of the average bit rate. We only consider luma PSNR because luminance channel is the most important one.

Table 1: Test Sequence Used

| Sequence | Resolution | Length (frames) | Source |
|---|---|---|---|
| SteamTrain | 2560x1600 | 250 | ITU |
| Netuba | 2560x1600 | 250 | ITU |
| Tears of Steel | 1920x1080 | 250 | mango.blender.org/download |
| Sintel | 1920x1080 | 300 | www.sintel.org/download |

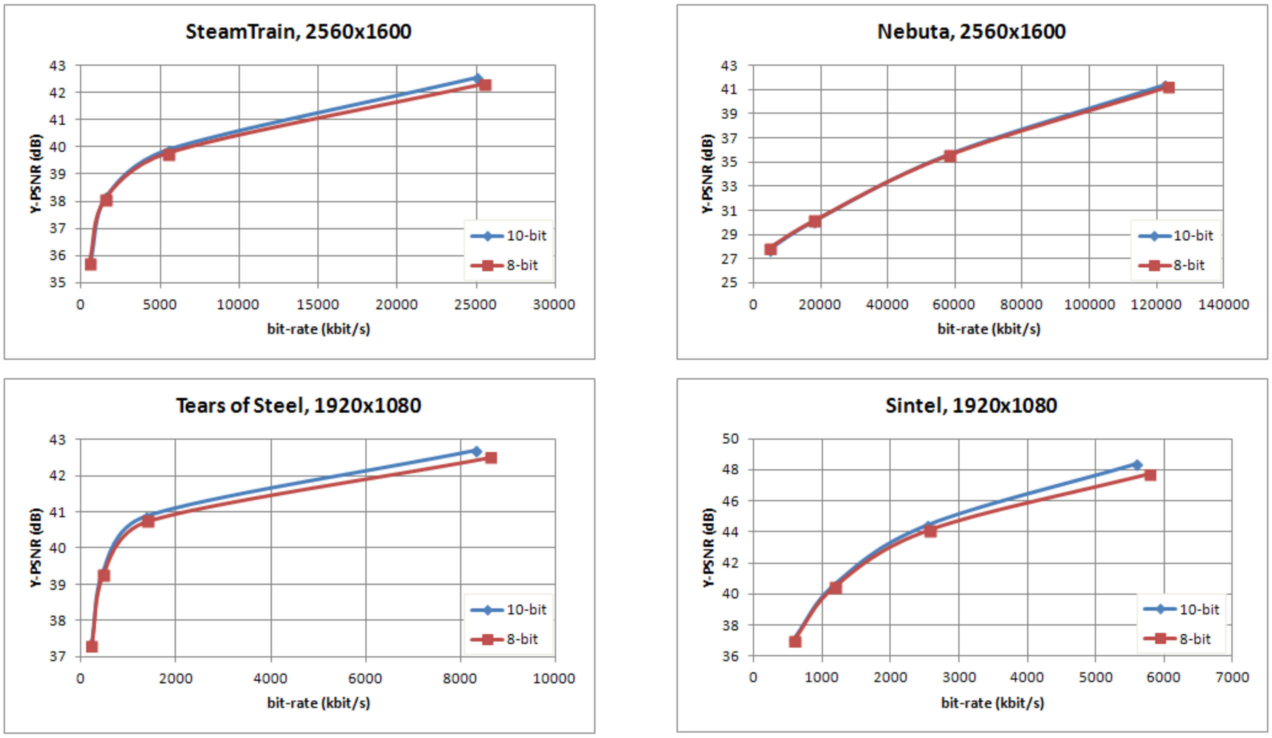


Fig. 2: Rate-distortion plots for test sequences

Results in Fig. 2 indicate that at lower QP and higher bit rate, 10-bit coding provides increased coding efficiency while maintaining the same quality. At higher QP and lower bit rate, the coding performance of 8-bit and 10-bit is similar.

The increased coding efficiency at higher bit rate was also reported in [5]. One of the reasons is that 10-bit pixels are better correlated than 8-bit pixels, and thus can be predicted more easily. This characteristic is well suited for transform based coding techniques such as AVC and HEVC. At low bit rate, on the other hand, the information in both 8-bit and 10-bit sources are heavily reduced and therefore the coding efficiency is about the same.

### Video Quality

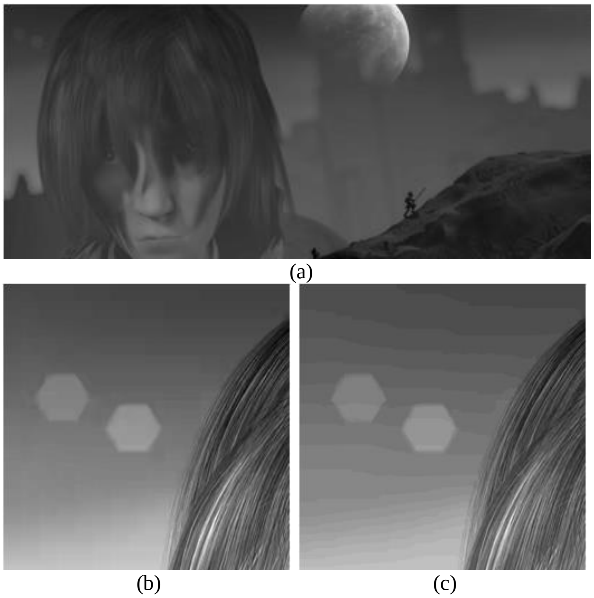
Evaluating objective video quality is a difficult subject. People often have different preferences when looking at video streams. Moreover, in our experiences, the PSNR difference between the 10-bit and 8-bit streams, which is less than 0.7dB, is not visible to most untrained eyes. Therefore, we judge video quality objectively using frame-by-frame comparison.

Fig. 3 shows the side-by-side comparison of a scene from the sequence “Sintel”. To show the quality of the video one would see on a 10-bit display, the following process is applied to the luma channel in order to generate this figure.

1) From the decoded 10-bit and 8-bit images, a sub frame of size 512x512 pixels is cut from the top left cover.

2) Pixels from the 10-bit image range from 138 to 278. They are normalized so that they map to 0 to 255. The output from this step is the 10-bit image Fig. 3(b).

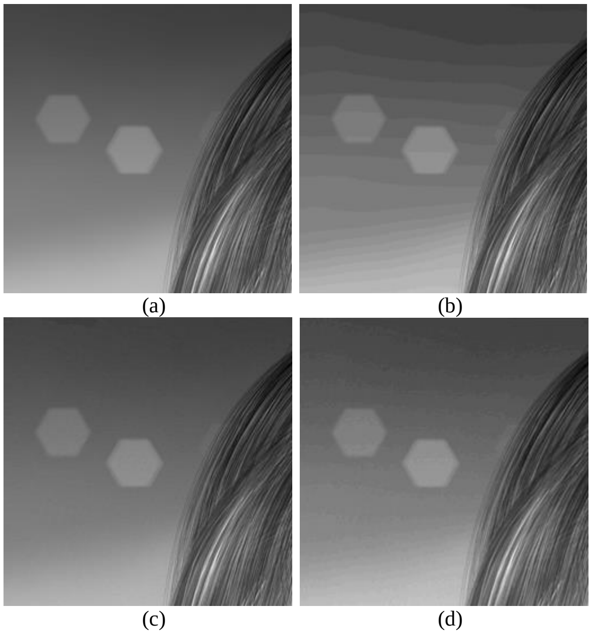
3) Pixels from the 8-bit image range from 34 to 70. They are also mapped from 0 to 255. The output from this step is the 8-bit image Fig. 3(c).



(a)

(b) (c)

Fig. 3: Side-by-side comparison of 10-bit and 8-bit images: (a) 8-bit original frame, (b) decoded 10-bit image, and (c) decoded 8-bit image.



(a) (b)

(c) (d)

Fig. 4: (a) 10-bit original “Sintel” frame, (b) 8-bit frame generated by rounding the LSBs, (c) 8-bit frame generated using Floyd-Steinberg dithering, (d) lowpass filtered frame in Fig. 4 (c).

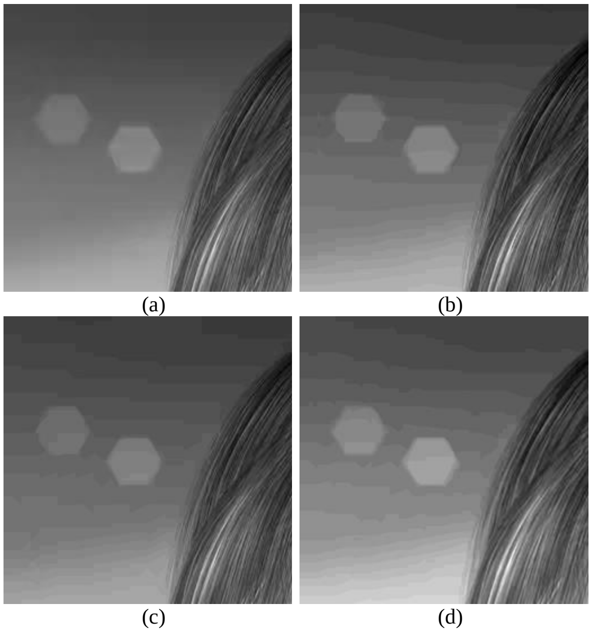
One can see that 10-bit compression removes contouring artifact noticeable in an 8-bit video. This is because 10-bit video can provide a higher dynamic range than that of 8 bits, and thus the transition from darker areas to brighter areas can be a lot smoother. This observation is also supported by [5] and [6].

## Contouring Countermeasures

It is generally accepted in compression labs that when a 10-bit or higher digital master exists, one way to reduce contouring is to dither the source 10-bit content before down sampling to 8 bits (Fig. 4 (c)). This is the current remedy widely used in the field. However this approach has two drawbacks.

1) The particular choice of the dithering algorithm introduces distortions to the original content and requires more human intervention in the encoding process.

2) In broadcast video, to hit compression targets, we often see noise reduction filters applied to images as part of the compression process. This noise reduction filter tends to undo much of the dithering introduced and would reintroduce contouring (Fig. 4 (d)).



(a) (b)

(c) (d)

Fig. 5: (a) decoded 10-bit “Sintel” frame, (b) decoded 8-bit frame generated by rounding the LSBs, (c) decoded 8-bit frame generated using Floyd-Steinberg dithering, (d) decoded noise filtered frame in Fig. 4 (c).

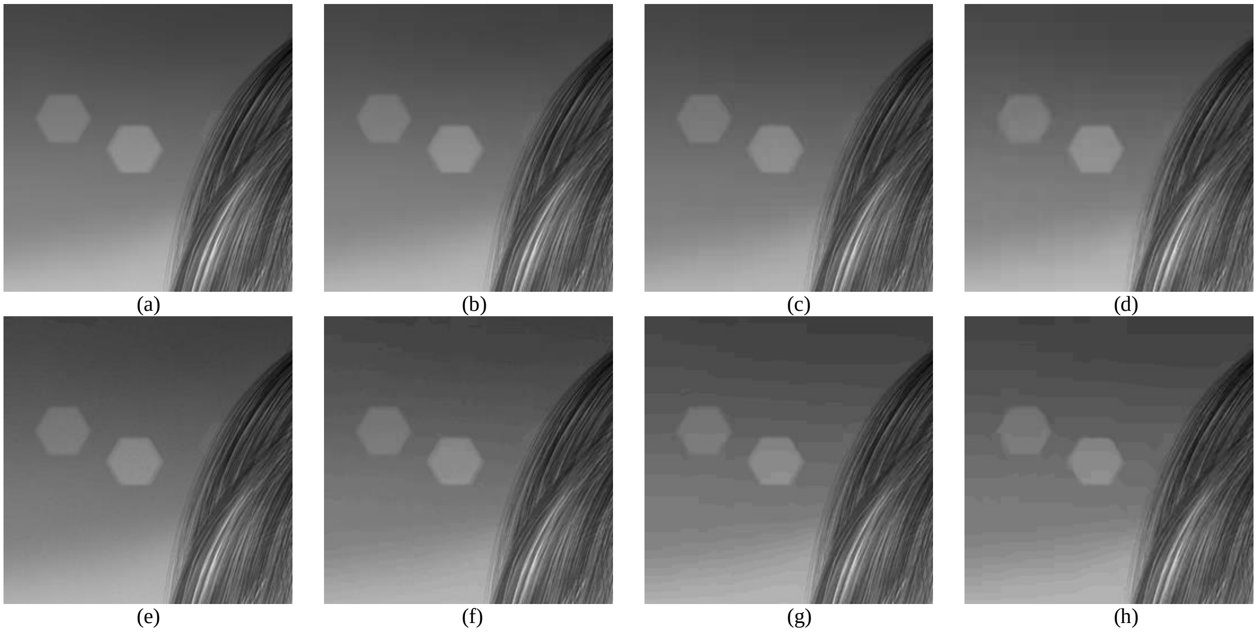

(a) (b) (c) (d)
(e) (f) (g) (h)

Fig. 6: (a) ~ (d) decoded 10-bit “Sintel” frame with QPs 5, 10, 15, and 20, (e) ~ (h) decoded dithered 8-bit “Sintel” frame with QPs 5, 10, 15 and 20.

Fig. 4 (a) shows a sub frame of the original 10-bit “Sintel” image. When a simple rounding method is used, banding is visible as seen in Fig. 4 (b). Fig. 4 (c) is an 8-bit frame generated from the original using Floyd-Steinberg dithering. Colour banding is not visible here. However, after noise filtering, traces of contouring reappeared in Fig. 4 (d).

Fig. 5 is the set of decoded frames from the original pictures in Fig. 4. The QP used to encode these frame is 22 for Fig. 5 (a), (b) and (c), which gives the final bit rate of the sequence of about 5.5Mbps. The lowpass filtered picture, Fig. 5 (d), can achieve similar bit rate with a slightly lower QP of 19. It is clear that contouring is visible in all the 8-bit images.

## Quantization and Contouring

Rate control in video compression (after considerations of tools used and within the same implementation) is largely determined by the quantization, which we call QP. We further publish our results to show how quantization values behave for both 10-bit and 8-bit video.

Fig. 6 shows the same decoded “Sintel” frame at QPs equal to 5, 10, 15, and 20 for both 10-bit and dithered 8-bit pictures. Banding starts to become visible for 8-bit compression when QP is 10, whereas it remains unnoticeable for 10-bit compression at QP equals 20. Given that a QP increase of 6 roughly halves the bit rate of a stream, this means that no artifacts are seen on the 10-bit sequence even though its bit rate is about a quarter of that of the 8-bit sequence. Even at very low bit rate, 500 Kbit/s and QP equal to 38, the 8-bit sequence still shows some contouring (Fig. 7 (b)). By comparison, the 10-bit sequence mostly exhibits blockiness, which is a common compression artifact at low bit rate (Fig. 7 (a)).

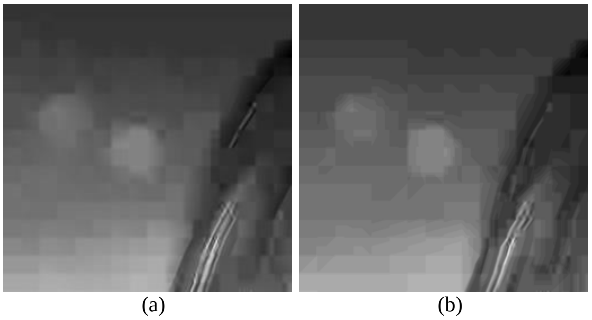

(a) (b)

Fig. 7: (a) decoded 10-bit “Sintel” frame at QP 38, (b) decoded 8-bit “Sintel” frame at QP 38.

## Memory Bandwidth Usage

The increase of coding efficiency and better video quality do not come for free. Implementing 10-bit video encoder and decoder has many difficulties. A critical subject in a video encoder or decoder design is memory bandwidth usage. This is because memory access is often the slowest part of a video coder pipeline due to the limit of the memory technology. In designing HEVC, some memory bandwidth considerations were put forward so that HEVC does not require much more bandwidth than AVC, even though HEVC uses an 8-tap interpolation filter for luma pixels [7]. Nevertheless, supporting a simplistic straight forward 10-bit video processing still requires memory bandwidth increase of about 25% over 8-bit (in theory). The caching of pixel blocks to avoid refetching previously fetched pixel data clearly can help, even though some judicious tuning is required to get more optimal cache management to make this work better.

One way to save on memory bandwidth is to reduce memory access when fetching reference pixels in motion compensation (MC). Since reference pixels usually exhibit strong spatial correlation, it is possible to compress the reference pixels before they are written to the frame buffer. Hence, memory bandwidth can be reduced for both memory read and write.

A simple lossless compression algorithm based on JPEG-LS is implemented in order to monitor memory bandwidth usage for an HEVC decoder [8]. MC cache is two-way set-associative with a size of 256x256 pixels.

Fig. 8 shows the bandwidth usage for the “Sintel” stream. Without reference picture compression, 10-bit stream requires about 21% more bandwidth than the 8-bit stream on average. *When compression is enabled, the 10-bit stream only needs about 4% more bandwidth on average than the 8bit stream.*

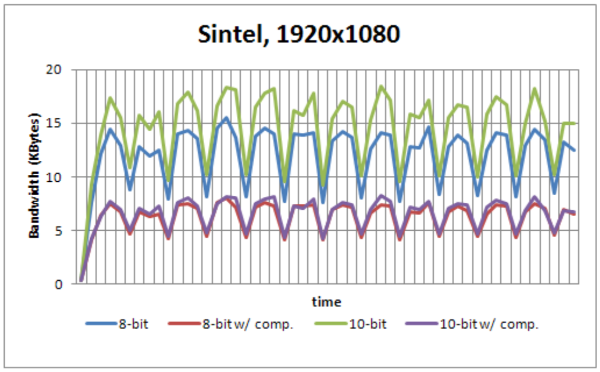


Fig. 8: Memory bandwidth measurement for 8-bit and 10-bit streams with and without reference picture compression

We also note that by turning on reference picture compression alone saves memory bandwidth usage by about 50% for the “Sintel” stream. It shows that this technique has great potential in supporting future codecs and profiles that include 12-bit and 14-bit video and YUV 4:2:2 and 4:4:4 formats.

It is understandable that bandwidth savings with compression is highly dependent on the content of the pictures. In the case of random noise, compression provides no savings at all. However, in most cases, bandwidth savings with compression is significant. We strongly suggest though that normal video for live content and animation are generally known to behave well with modern compression algorithms, since such codecs were targeted at a wealth of such content and improvements were designed and improved upon the successes of behavior with previous algorithms.

Most significant finding here is that we can support lossless 10-bit frame buffer compression with an incremental memory bandwidth increase of 4% over 8-bit frame buffer compression (when the raw traditional approach would require 25% more memory bandwidth). This is clear indication to us that supporting higher dynamic range video should strongly consider frame buffer compression as a key architectural focus of building IC codecs for both encode and decode. The result will carry over to 12bit implementations as well.

## CONCLUSIONS

In this paper, we have discussed various aspect of 10-bit video processing. We have shown that comparing to 8-bit video, 10-bit has increased coding efficiency because 10-bit pixels are well suited for transform based video coding techniques. Furthermore, because of its higher dynamic range, 10-bit video can reduce banding or contouring artifacts visible in an 8-bit video, especially in areas with low chrominance variation. We have examined one particular aspect of 10-bit encoder and decoder implementation, namely memory bandwidth usage. By using reference picture compression, we are able to limit bandwidth increase because of 10-bit to about 4%. We believe that with minimal cost increase, broadcasters should be able to deliver 10-bit content and provide visible quality improvement to the consumers.

## REFERENCES

[1] D. B. Judd and G. Wyszecki, *Colour in Business, Science and Industry*. Wiley Series in Pure and Applied Optics (3rd ed.). New York: Wiley-Interscience, 1975

[2] http://reddoorreview.com/house-of-cards-review/

[3] B. Bross, W.-J. Han, G. J. Sullivan, J.-R. Ohm, and T. Wiegand, *High Efficiency Video Coding (HEVC) Text Specification Draft 10*, document JCTVC-L1003, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC). Jan. 2013.

[4] K. McCann, B. Bross, W.-J. Han, I. K. Kim, K. Sugimoto, G. J. Sullivan, *High Efficiency Video Coding (HEVC) Test Model 12 (HM 12) Encoder Description*, document JCTVC-N1002, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC). Aug. 2013.

[5] P. Larbier, *Keep Video Quality Pristine throughout the Prodcution Process: 4:2:2 10-bit AVC/H.264 encoding*, http://x264.nl/x264/10bit_03-422_10_bit_pristine_video_quality.pdf

[6] A. Duenas, A. Malamy et al, *On a 10-bit consumer-oriented profile in High Efficiency Video Coding (HEVC)*, document JCTVC-K0109, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC). Oct. 2012.

[7] T. Hellman, W. Wan, *Motion Compensation Restrictions to Alleviate Memory Bandwidth Concerns for High Resolution Video*, document JCTVC-H0441, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC). Jan. 2012.

[8] M. J. Weinberger and G. Seroussi, *The LOCO-I Lossless Image Compression Algorithm: Principles and Standardization into JPEG-LS*, Proceedings of the IEEE Data Compression Conference, Snowbird, Utah, Apr. 1996.