Discussion regarding S-FIVE Bonus CE

The issue

During the manipulation of some videos containing very dark or very bright visual content, significantly different results were found to be obtainable when simple adjustment of levels were executed.

Results appeared to be dependant on the software used. The adjustments were made in an attempt to reveal detail which was not previously visible to the human eye - being at the extremes of the eye's sensitivity to light e.g. using brightness/contrast/gamma adjustments. The issue only appeared to manifest for some such videos, and not others.

When the phenomena was observed for a particular video, it manifested as an absence of detail below a certain threshold in the dark areas of imagery, which was otherwise demonstrably present if the footage was manipulated in other software. Likewise in light areas, detail above a certain threshold would similarly appear to be absent, and unrecoverable.

In summary, it appeared that detail at the extremes of dark and light were being clipped depending on what software was used. This raises concerns that an examiner may not realize that they are not able to access all the data present in a video file if they are using inappropriate software.

What causes this issue?

Typically, compressed digital video is stored in the *Y,Cb,Cr* colour space as Luminance (*Y*) and two colour difference components (*Cb* and *Cr*). These colour components are commonly subsampled to save bandwidth.

By subsampling the colour components, the *Y,Cb,Cr* colour effectively mimics the human vision system in terms of prioritizing capture of a-chromatic detail (from *rod* sensors in the retina) over chromatic detail (from the *cones*). Encoding digital video in this form therefore theoretically allows significant compression to be achieved (with low perceptual loss of detail) compared to *RGB* encoded video - which requires the same resolution to be stored for each of the 3 colour channels. Video encoded in the *Y,Cb,Cr* colour space will be identified with the property *'YUV'* or *'yuv'*.

Commonly digital video is recorded as *'YUV420P'.* This means that it has been recorded in the *Y,Cb,Cr* colour space, with 4:2:0 chroma subsampling. The *'P'* refers to the planar structure of the data format, in as much as the three planes are stored as sequential arrays of data. *YUV420P* encoded video occupies half the bandwidth of *RGB* encoded video.

The screen of a television/monitor works in the *RGB* domain. Therefore Red, Green and Blue colour channels must be reconstructed from the luminance (*Y*) and subsampled Blue (*Cb*) and Red (*Cr*) values in order to display *Y,Cb,Cr* encoded video.

It should be noted that conversion from *RGB* to *Y,Cb,Cr* and back to *RGB* again is a lossy process due to sub-sampling (if applicable) and rounding errors.

The issue which is the subject of this document arises because there are two different standards in use for converting *Y,Cb,Cr* values to *RGB*:

- The convention is that **video** *Y,Cb,Cr* should use byte values ranging 16…235 for luminance (220 discreet values between black…white), and 16…240 (225 values) for chrominance. This is known as '*limited range*' or '*mpeg range*'.

- Conversely, the convention for **jpeg still images** is to use the *full* 0…255 range in all planes (256 discreet values black…white). This is known as *'full range'* or *'jpeg range'*.

Not all manufacturers of recording equipment adhere to these standards, and some choose to use the ***'full range'*** *Y,Cb,Cr* when recording **video footage** - possibly to allow for improved signal to noise ratio. However, some vendors are incorrectly flagging the colour space range within their video, and some are neglecting to flag the range at all.

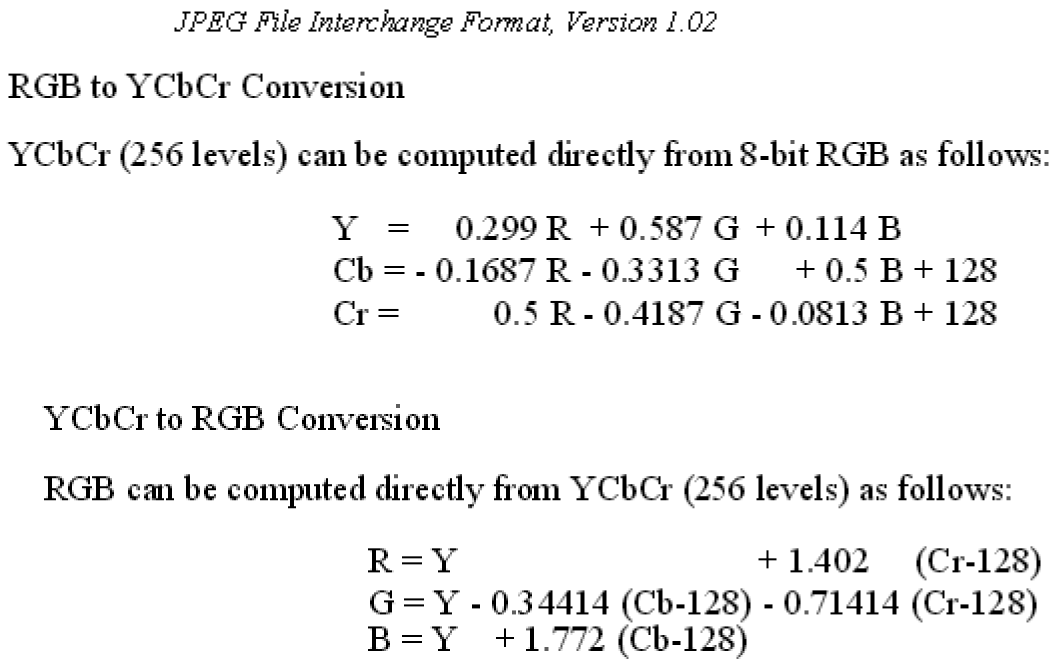
The issue, discussed in the first section of this paper, arises due to **video** which has been encoded using the ***'full range'*** of the *Y,Cb,Cr* colour space being converted incorrectly by software when opening the file - this can be due to the flag within the video file being either absent, set incorrectly, or due to a correctly set flag not being read by the enhancement software.

If *'full range'* video is miss-identified as '*limited range'* video, then the values at the extremities of the 'full range' will be clipped in the conversion process - effectively expanding what is expected to be a reduced range of 16-235 across the entire 0-255 range for each colour channel in the *RGB* space.

Colour Space Conversions

Figure 1 contains an excerpt from the JFIF specification defining the coefficients for *Y,Cb,Cr* (*Full Range*) <-> *RGB* conversion:

Figure 1: JFIF v1.02 (0...255)



JPEG File Interchange Format, Version 1.02

RGB to YCbCr Conversion

YCbCr (256 levels) can be computed directly from 8-bit RGB as follows:

$$\begin{aligned} Y &= 0.299\,R + 0.587\,G + 0.114\,B \\ Cb &= -0.1687\,R - 0.3313\,G + 0.5\,B + 128 \\ Cr &= 0.5\,R - 0.4187\,G - 0.0813\,B + 128 \end{aligned}$$

YCbCr to RGB Conversion

RGB can be computed directly from YCbCr (256 levels) as follows:

$$\begin{aligned} R &= Y + 1.402\,(Cr-128) \\ G &= Y - 0.34414\,(Cb-128) - 0.71414\,(Cr-128) \\ B &= Y + 1.772\,(Cb-128) \end{aligned}$$

The BT.601 ITU specification contains information about recommended colour space for **video**.

In Figure 2, an approximation of the relationship between '*Full Range'* Y and '*Limited Range'* Y, derived from reading the BT.601 ITU specification is provided:

Figure 2:

Full Range Y is mapped to *Limited Range Y* via: 16+(*Y*-0)*(219/255) = *Limited Range Y*
Limited Range Y is mapped to *Full Range Y* via*:* 0+(*Y*-16)*(255/219) = *Full Range Y*

Flagging of colour space range in .h264 video

In *h.264*, an optional part of the bitstream data is called the *VUI* (as defined in ISO/IEC14496-10), when utilized this can contain a single bit flag called *video_full_range_flag*, enabling this flag tells the decoder to process as '*full range'* (0…255). The default is to assume the '*limited range'* (16…235).

In sample data, some videos have been found to contain the flag, explicitly specifying '*limited range'* when in fact the *'full range'* was in use, some videos where the flag was correctly used and others where it was absent, thereby incorrectly implying the default '*limited range'*.

Further work needs to be undertaken to identify which codecs/containers allow for the flagging of the *Y,Cb,Cr* colour space range.

The Collaborative Exercise:

Download the Collaborative exercise clip 'MOV01469.MPG' from the following webpage:

https://www.s-five.eu/mediawiki/index.php/Bonus_CE

Download FFmpeg from the following webpage:

http://ffmpeg.zeranoe.com/builds/

1. Firstly, using *ffmpeg.exe*, lets convert the file 'MOV01469.MPG' into a series of bitmap stills.

In a command window, type the following code:

```
"folderlocation\ffmpeg.exe" -i "folderlocation\MOV01469.MPG" -vsync drop -f image2
-pix_fmt bgr24 "folderlocation\filename%07d.bmp"
```

Substitute the green text for the pathway of the relevant folder location (e.g. C:\documents and settings\user\My Computer\ffmpeg\ffmpeg.exe

This will convert the video file into a series of bitmaps: 'filename0000001.bmp', 'filename0000002.bmp', 'filename0000003.bmp'…etc (The no. of zeros is specified by '%07d'). As we have not specified how to handle the conversion from *YUV420P* to *RGB*, FFmpeg will as default perform the conversion interpreting a 'limited range' from the source clip.

In any picture editor, try adjusting the gamma of the first output frame. You will find that there is no detail present in the blacks.

2. Next, using *ffmpeg.exe*, we will perform the same conversion as in step1, but this time specifying that the original file contains *'full range'* values, rather than *'limited range'*. Take care to use a new output folder.

In a command window, type the following code:

```
"folderlocation\ffmpeg.exe" -i "folderlocation \MOV01469.MPG" -vsync drop -f image2
-pix_fmt bgr24 -vf scale=in_range=full:out_range=full "folderlocation
\filename%07d.bmp"
```

Substitute the green text for the pathway of the relevant folder location (e.g. C:\documents and settings\user\My Computer\ffmpeg\ffmpeg.exe

This time we have specified how to handle the conversion from *YUV420p* to *RGB* using the code: '-pix_fmt bgr24 -vf scale=in_range=full:out_range=full'. We have instructed FFmpeg to interpret the input video file as 'full range', and to output also as *'full range'* using a 24bit *RGB* colour space (specifically *BGR24*).

In any picture editor, try adjusting the gamma of the first output frame. You will now find that there is much more detail available in the blacks.

Many of those who carried out the Bonus Collaborative Exercise found that some software (for Example VLC Media Player) correctly dealt with the example video file, and allowed adjustment of levels, brightness & contrast & gamma to reveal the detail in the blacks. This may be because the filters operate in the YUV domain, and hence the enhancement takes place prior to the conversion to *RGB*.

Experimentation

Method

In order to diagnose the issue, low light video was collected from a variety of different format devices. The video was subsequently filtered/color space converted/transcoded using a variety of software packages and in different combinations to isolate where and why the issue was occurring.

A coarse way of seeing when the issue was present quickly became apparent:

Stage a: Exporting a *BGR24* bitmap uncompressed video using FFmpeg.
Stage b: A gamma adjustment of the exported *BGR24* video, from stage a, using FFmpeg.
Stage c: A gamma adjustment of the original video using FFmpeg.

The video from stage b could then be compared against that produced by implementing the gamma adjustment filter on the original video during stage c. Clips affected by the issue would contain much more detail in the low light areas when using stage c filtering on the original.

Results

The use of the *'full range' Y,Cb,Cr* was found to be widespread throughout a variety of different formats. Older model video cameras recording in *mpeg-1* were found to be as likely to implement it as were proprietary CCTV and *h.264* generated by modern phones.

Regarding phones: apps on a given device were found to determine whether the video would be affected rather than the make/model of phone itself. This is an important distinction as it means there is no gain in making an affected device list to improve image handling.

Processing video that has been recorded using the '*full range'* mapping as '*limited range'* will lead to all values outside 16…235 effectively being clipped. Everything from 0-15 will be set to the same level of black, everything from 236-255 will be set to the same level of white. This is especially noticeable in the darker range where a lot of detail can be held in low light situations.

The following images demonstrate the clipping that can occur. The first set of three images (Figures 3, 4, 5) show the effect of processing the example after converting to *BGR24* using the default '*limited range'*. The second set of three images (Figures 6, 7, 8) show the difference when converted to the *BGR24* pixel format by specifically choosing to use the '*full range'*.

The first waveform in Figure 3 demonstrates that values below the *'limited range'* level of 100% black (i.e. with a byte value of 16) will be flattened to saturated black when improperly converted.

The images were generated using FFmpeg, as described in the collaborative exercise section, to export a *BGR24* bitmap image from a *YUV420P* source video recorded using the *'full range'*. There is no bitstream flag to tell the decoder that this is the case so the default FFmpeg export contains clipped data. When set explicitly to map from a *'full range'* stream, the colour space conversion occurs correctly and transfers all the expected values.

Forcing to '*full range'* was achieved by processing in FFmpeg using the following instruction when converting colour space: “-vf scale=in_range=full:out_range=full”

Figure 3: YUV Waveform from test clip (clamped to demonstrate clipping)

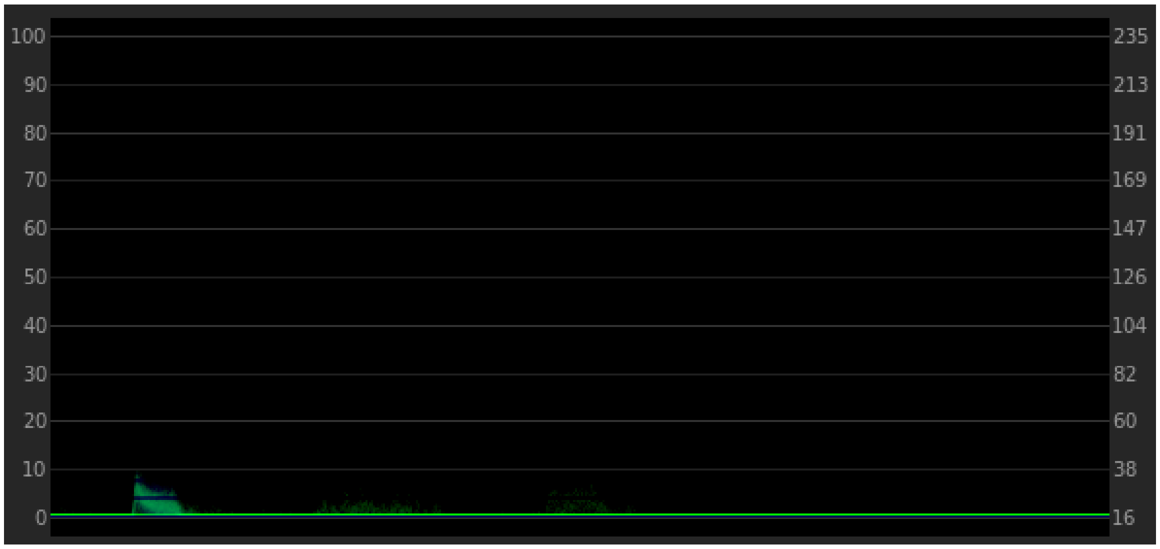


Figure 4: Test clip using default export to BGR24 bitmap (no gamma shift)

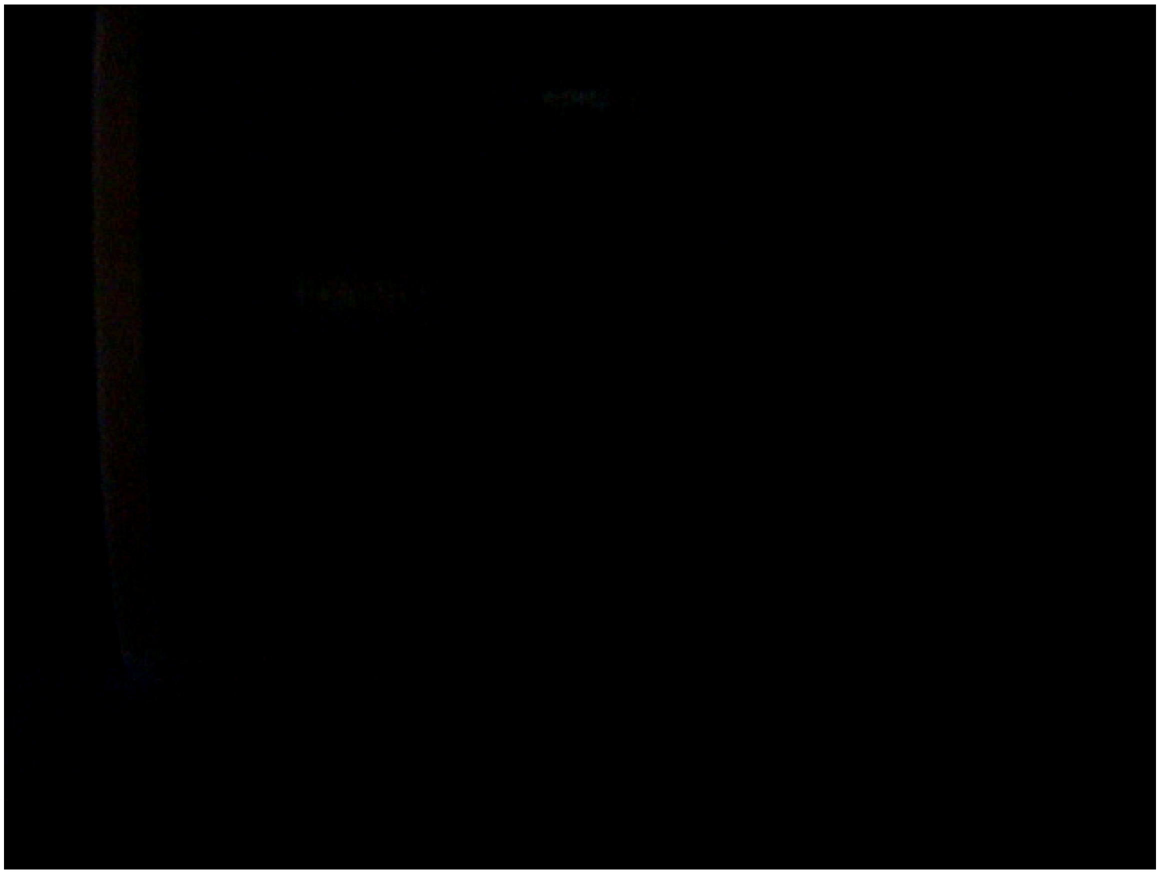


Figure 5: Test clip using default export to BGR24 bitmap (4.0 gamma shift in Irfanview)

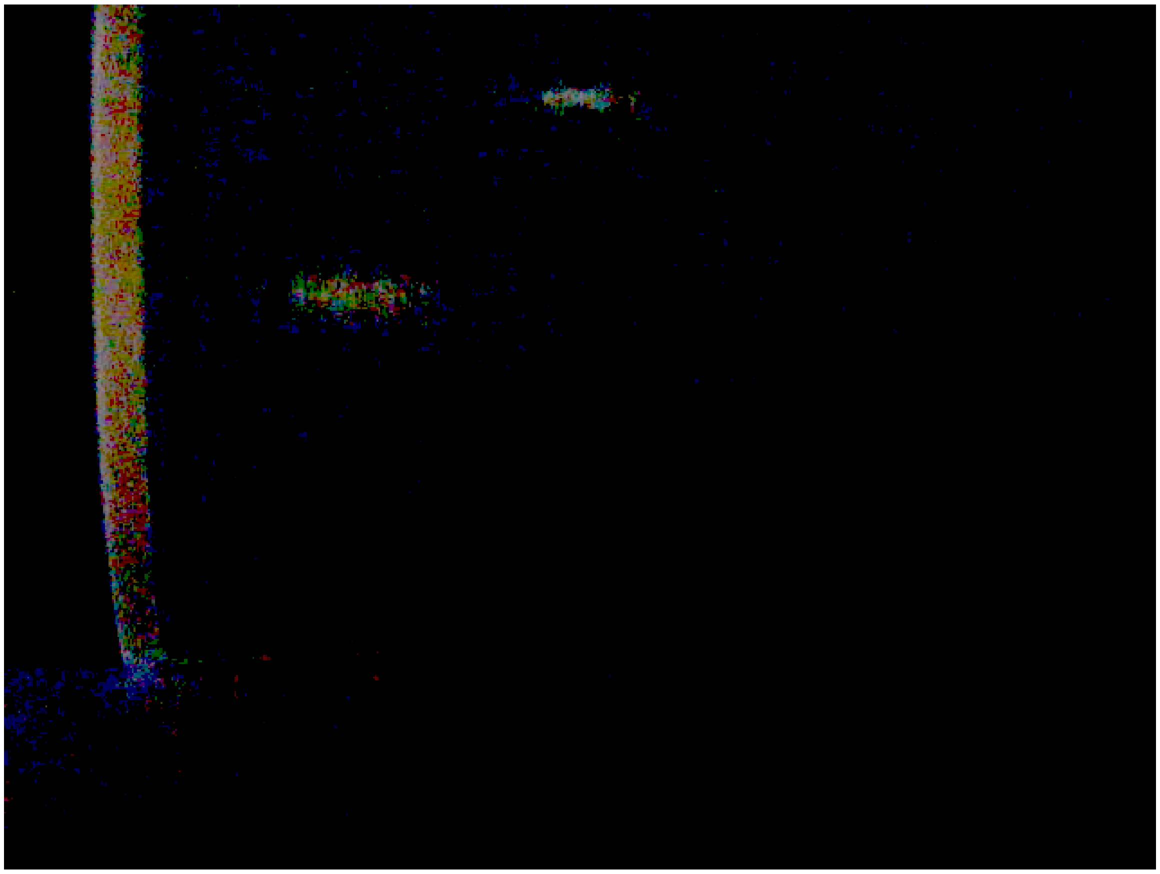


Figure 6: YUV Waveform from test clip

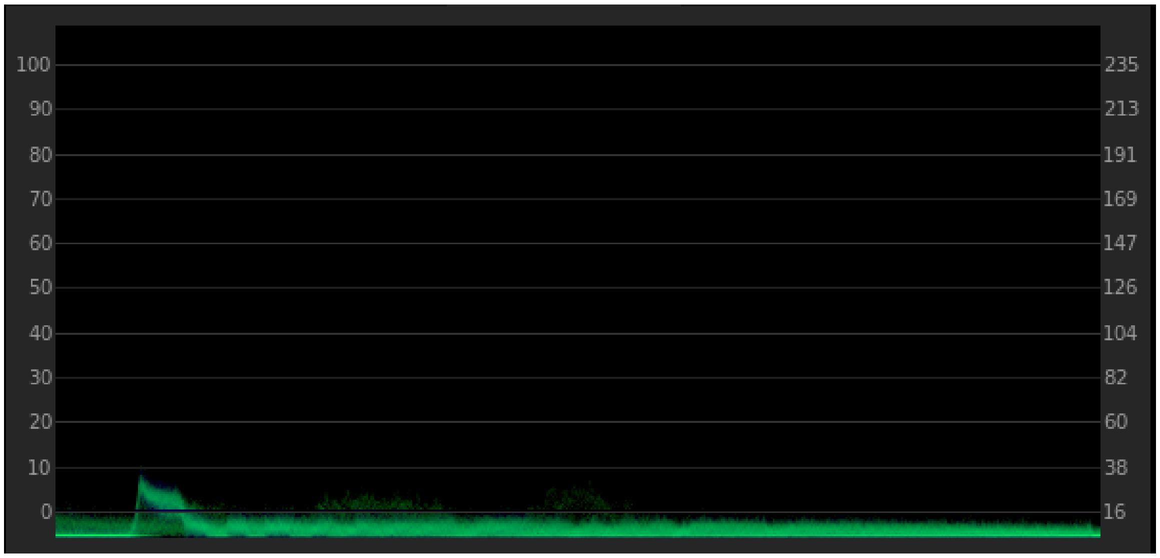
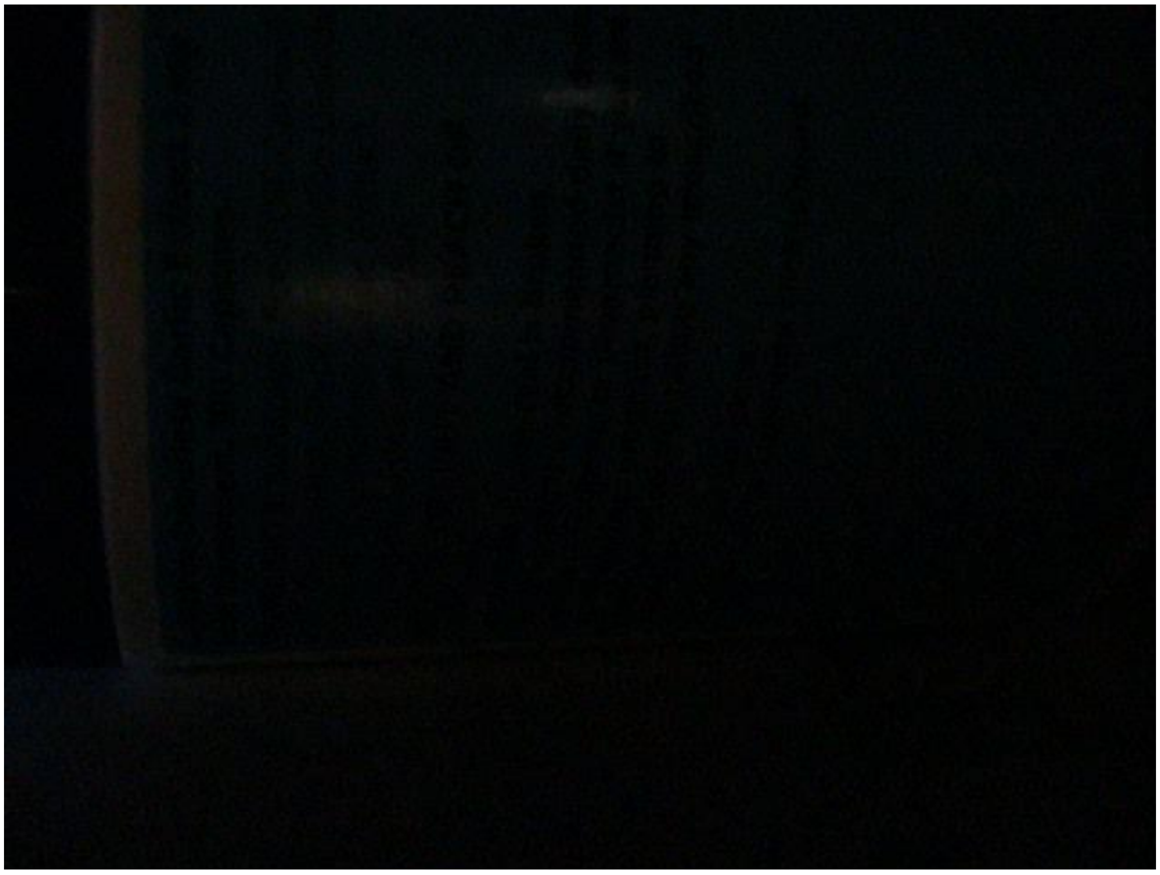
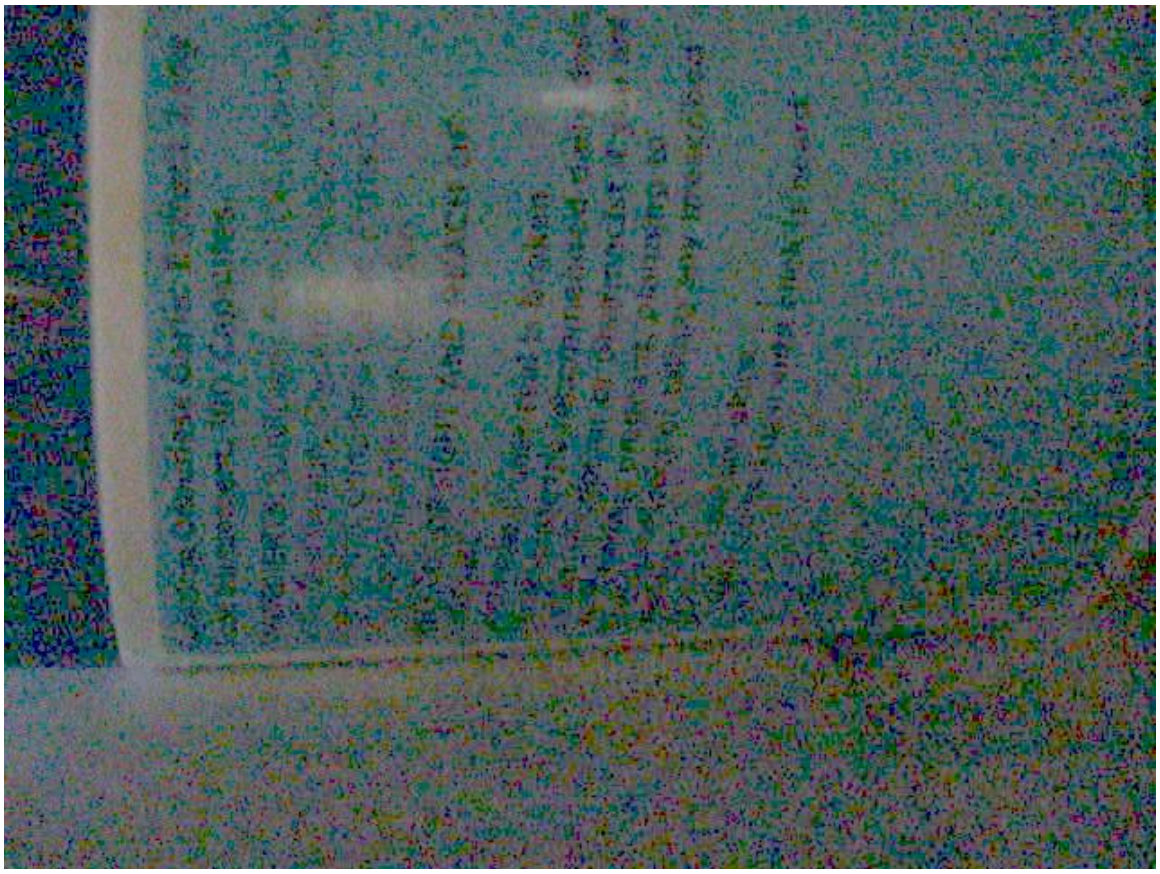


Figure 7: Test clip using forced full range export to BGR24 bitmap (no gamma shift)



<u>Figure 8: Test clip using forced full range export to BGR24 bitmap (4.0 gamma shift in Irfanview)</u>



Conclusion

Ideally, where possible, light levels should either be processed using *YCbCr* filters, or converted to *RGB* using awareness of the correct colour space range.

Filtering/conversion of video media can lead to unpredictable results. VLC for example processes its gamma shift in the *YCbCr* domain meaning that the range is irrelevant when filtering - it is only when converted to *RGB,* whilst being rendered to the screen/exporting as stills, that the *out of range* values become clipped.

Unfortunately the majority of applications do not document the colour domain in which their filters operate. Likewise there is no guarantee that all the filters in a given application work in the same colour space. It should also be noted that a video acquired through screen capture will be ingested in the *RGB* colour space, therefore any clipping that has occurred prior to being rendered to the screen will subsequently be unrecoverable on the captured output.

The suggested best practice is, where at all possible, to diagnose a given file in the *YCbCr* domain via a waveform monitor, to assess the max/min *Y* values. Alternatively, values can be parsed by a stream analysis tool looking at the raw uncompressed *YCbCr* data. A preprocessing stage is necessary as the bitstream metadata cannot be trusted to have been accurately implemented. Likewise the *YCbCr* implementation is not hardware specific and may occur in either form on a given device i.e. each clip must be checked in isolation rather than assuming all videos from that device will be the same.

It should be mentioned that processing a *'limited range'* clip as '*full range'* is not as detrimental compared to processing *'full range'* as *'limited range'*.

During testing, it was also observed that files appearing to have been created using the 'limited range' often stray outside the 16…235 range - this has been noted as being due to noise and the tolerance of the DSP producing the video, post processing such as overlays can also be a contributor.

Further research

Which formats/containers contain the ability to flag the colour range?

What colour spaces are being used by the filters/transforms of mainstream video encoding/editing/enhancement software?

How much influence do graphics cards have on the process?

Truncation of values - *YCbCr* dynamic range contains a wider gamut than RGB, making some *YCbCr* values illegal in the RGB domain. How does this relate to forensic handling of material? Is the broadcast conversion technique for out of gamut values compatible with forensic needs regarding handling of clipped/out of range values?

Different subsampling techniques appear to lead to different artifacts being generated at encode/decode, is there a best practice for forensic reconstruction of subsampled images?

Further reading

http://www.poynton.com/PDFs/coloureq.pdf
http://www.poynton.com/PDFs/ColorFAQ.pdf
http://www.w3.org/Graphics/JPEG/jfif3.pdf
https://www.itu.int/rec/R-REC-BT.601/en
https://www.itu.int/rec/R-REC-BT.709/en
http://www.compression.ru/download/articles/color_space/ch03.pdf
http://downloads.bbc.co.uk/rd/pubs/reports/1987-22.pdf
http://www.glennchan.info/articles/technical/chroma/chroma1.htm
http://stnsoft.com/DVD/color_pick.html
http://www.intersil.com/content/dam/Intersil/documents/an97/an9717.pdf
https://ffmpeg.org/doxygen/2.1/yuv2rgb_8c_source.html
https://msdn.microsoft.com/en-us/library/ms893078.aspx

Acknowledgements

Project S-FIVE is an ENFSI Monopoly Programme, financed by the European ISEC Programme: Prevention of and fight Against Crime Action Grants;
European Commission - Directorate-General Home Affairs.

More information about the S-FIVE project is available at:
http://s-five.eu/
or through the ENFSI Digital Imaging Working Group (DIWG);
http://ensfi.eu/about-enfsi/structure/working-groups/digital-imaging

The S-FIVE Project Team would like to thank Gareth Harbord and Simon C. Latham, Metropolitan Police Service, London, UK, for their work on composing the S-FIVE Bonus CE and this document.