Ari Greenberg
12/11/19
NYU-MIAP Fall 2019
Digital Preservation
Prof. Nicole Martin

Color Me Confused:
Color Space Preservation Challenges
in the Transcoding, Rewrapping, and Playback of Digital Video Files

I. Background/History

The fixity of color in digital video files remains a difficult challenge not only for audiovisual archivists, preservationists, and conservators, but even for those working in the "production pipeline" as well. While color timing and correction are almost as old as audiovisual media itself, the ongoing and ever-widening supplanting of analog media with digital counterparts, while liberating in some regards, has in others made color management more impenetrable than ever before. Managers and technicians working with audiovisual collections must forge a middle path between audio and video engineers on the one hand— those actually responsible for designing new algorithms and enshrining new technical standards and specifications— and their non-audiovisual-specialist colleagues— and the general public— on the other. The terms *codec* and *container* remain opaque, even as screens of all sizes (and of steadily growing color gamuts and resolutions) proliferate.

Even with a basic understanding of color science and digital asset management, ensuring color fixity, backwards-compatibility, and interoperability in general, remains a challenge. Archivists will frequently transcode or rewrap a video file — altering either the container, or the codec itself— only to find the color has perceptibly shifted, usually appearing either over- or under-saturated. Diagnosing the problem can prove vexing, even for seasoned specialists. Color, in short, remains a mystery.

In part, this mysteriousness arises out of the confusing and inconsistent terminology used by those in the field. As Andrew Oran and Vince Roth note in their brief primer, "Color Space Basics," "the terms Color Model and Color System— as well as Color Space— are often used interchangeably."[1] Recommendation standards such as BT.709 and BT.2020 are referred to as "color spaces" even though they represent a wider "range of specifications," beyond color alone.[2]

Nonetheless, Oran and Roth draw some distinctions for the sake of clarity: *Color space* properly refers to "[a] combination of a color model *plus* a precise description of how the colors within that model are to be mapped in 3D space, allowing for specific colors to be identified by coordinates within a given color space." They define *color model* as "[a] mathematical representation of color, based on 3 or 4 values (i.e., RGB or CMY and luminance, often represented as Y or K)"; and *color system* as synonymous with *color model.* *Color gamut*, it might be added, describes the range of color visible to the human eye that a given standard is able to capture, based on the so-called CIE color space diagram, issued in 1931 by the *Commission internationale de l'éclairage* (International Commission on Illumination). *Bit depth* refers to the number of bits required to describe one pixel in the file— a higher bit depth, therefore, indicates a higher color gamut.[3]

---

[1] https://amianet.org/wp-content/uploads/Publication-AMIA-Tech-Review-V4-2012.pdf

[2] https://www.arri.com/en/learn-help/learn-help-camera-system/camera-workflow/image-science/frequently-asked-questions-on-hdr#accordion-44200

[3] https://www.nyu.edu/tisch/preservation/program/student_work/2017fall/17f_1807_Dowlatshahi_a2_paper.pdf

All this is arcane enough. However, to complicate matters further, the various technical recommendations of the International Telecommunication Union Radiocommunication Sector — having been issued now for decades— are also referred to as *color spaces*, even though, as will become evident in this paper, they in fact usually entail much more than color space alone. Furthermore, discrete attributes of video files that an archivist might assume are always paired do not necessarily have the one-to-one relationship with color information that they might seem to at first glance. For instance, says one archivist, "some would take a BT.601 720x480 and scale it up to a BT.709 1920x1080 and leave the color just as it is, while others may include a BT.601 to BT.709 transform along with the frame size scale."[4] The bottom line is: there is room for interpretation, but documentation is key.

Not unrelatedly, it is worth acknowledging here from the outset that it is a somewhat arbitrary (and essentially impossible) exercise to attempt to discuss (in a vacuum) issues around color space in rewrapping and transcoding without raising other issues of audiovisual data migration and fixity more generally. To apply metaphorically a quote from the ecologist Garrett Hardin, when it comes to rewrapping and transcoding digital video files, "You can never merely do one thing."[5] This sentiment is as applicable to the digital ecosystem as it is to the natural one.

This paper focuses on three ITU-R specifications: BT.601, BT.709, and BT.2020. While the first two came onto the scene during the analog era, they have nevertheless carried over into

---

[4] Email with Dave Rice.

[5] https://archive.org/details/environmentalsci00mill_0

the digital age, and their backwards compatibility with systems using BT.2020 (and other newer specifications) remains an issue for archivists.

BT.601 was issued by the ITU in 1982 for use with cathode ray (CRT) monitors, using standard definition (SD) interlaced video. BT.709 was issued in 1990 for high definition (HD), reproducing roughly 34% of the human color gamut. BT.709 is currently the industry standard for HD video. BT.2020 was first issued in 2012 and constituted an approximately 30% increase over its predecessor. The most recent standard, BT.2100, uses the same color space as BT.2020. BT.2020 and BT.2100 are both referred to as Wide Color Gamut (WCG) color space specifications, an umbrella term encompassing all new specifications after BT.709.[6]

II. Preservation Challenges

The challenges preservationists face vis-a-vis color spaces are threefold:

- Color information itself (the codec, also known as the stream, or essence) becoming compromised following transcoding;
- Interoperability issues following rewrapping (changing the container) as a result of insufficient or inaccurate self-description, including but not limited to color metadata;
- and, presentation inconsistency (in which the file exhibits color shift when played back on one or more given players in a workflow).

For this paper, several archivists shared unresolved color fixity issues they had recently encountered within their respective workflows.

---

[6] https://www.arri.com/en/learn-help/learn-help-camera-system/camera-workflow/image-science/frequently-asked-questions-on-hdr#accordion-44200

First Scenario: Color Shift in an Access File (H.264)-Creation Workflow

In the first scenario, in the course of creating H.264 web access copies of ProRes 4:2:2 HQ files using ffmpeg using the simple "Transcode to H.264" script provided by the script repository ffmprovisr, one archivist found that, despite successfully transcoding the file, upon viewing the transcoded image, it was clear that the color had become desaturated in the new access copy. As a troubleshooting measure, the transcode was repeated, this time with one alteration: the chroma subsampling pixel format was now specified in the script as yuv422. Nonetheless, once again, the colors appeared desaturated in the transcoded file. The origin of the problem remained unclear— whether the desaturation effect originated in the ffmpeg script on the one hand, or, rather, in the playback software or the monitoring environment on the other. Finally, the original ffmprovisr-recommended script was retried, but this time omitting the `-pix_fmt` flag. According to the documentation available at ffmpeg.org, this flag, a command to set the pixel format in the transcode, will "select the best pixel format supported by the encoder" in the event that "the selected pixel format cannot be selected."[7]

Apparently, while the application of the yuv422p flag should have tipped the player off to the fact that the file had a sampling frequency of 4:2:2, it had nevertheless persisted in expecting a sampling frequency of 4:2:0 with the `pix_fmt` flag included; once removed, "the transcode looked way better and the [pixel] format defaulted to yuv422." This is what audiovisual archivist Dave Rice refers to as the phenomenon by which digital video files are "at risk of interoperability issues as [they depend] on adherence to presumptions rather than clear

---

[7] https://ffmpeg.org/ffmpeg.html

self-description."[8] In this case, the player "presumed" a sampling rate of 4:2:0, until tinkering with the ffmpeg script set it straight. Unfortunately, even with this course correction, some desaturation persisted in the image, with the cause remaining undiagnosed.

In another scenario, archivist Shahed Dowlatshahi at California Revealed was similarly involved in a workflow creating H.264 MOV access files from DPX 2K preservation master files. While the same access file presented with the desired colors in VLC and QuickTime, in Adobe Premiere the same file exhibited signs of desaturation. One hypothesis postulated that Dowlatshahi's version of Premiere was not up-to-date, and therefore could only display BT.709. Nonetheless, one could easily make the case that "it's a balance between interoperability and accuracy, and BT.2020 is more efficient."[9]

A Google search reveals that this is not an uncommon problem for those in the field working on color management with Adobe Premiere. Until the October 2018 update of Premiere Pro CC (Version 13), an archivist would apparently have had to import the preservation master into Adobe After Effects and "render out" the embedded color profile.[10] However, following the 2018 update, Premiere's preferences include an *Enable Display Color Management* switch.

But the "switch" is not a perfect solution, because— as noted briefly above— the issue of monitor environment interoperability remains an issue even when the playback software is able to correct for BT.2020 on a system built for BT.709. In fact, the monitor environment issue remains even when BT.2020 is *not* involved, since "true" BT.709 will never display correctly on

---

[8] https://www.tate.org.uk/about-us/projects/pericles/audiovisual-adherence

[9] Email correspondence with Dave Rice, Nov.2019.

[10] https://community.adobe.com/t5/premiere-pro/rec-2020-color-space/td-p/9157515

certain widely-used monitors, such as an sRGB (standard Red Green Blue) display. Designed by Microsoft and HP in 1996, the sRGB color space, while the "default" on many consumer-grade monitors, is nevertheless hardly a one-size-fits-all solution. Since the architecture of most sRBG monitors is 8-bit, BT.709 in Premiere on an sRGB monitor will appear desaturated with the "Display Color Management" option selected when played back through a web application (YouTube, Vimeo, and so on), but will appear "true" when played back by a broadcaster.[11]

The archival consensus in any event names insufficient self-description as the central problem in maintaining color space fixity. As Rice notes, an archivist can identify a file's format fairly easily, but "answering questions such as 'Is this sufficiently self-descriptive?' or 'Why doesn't this file work with this software?' require delving deeper."[12] Rice also notes that, contrary to popular belief, only a small subset of uncompressed video is highly interoperable— again, largely due to the fact that it relies on its container to provide description.

III. Testing for color fixity with MediaInfo and QCTools

First Test: DV → MOV

As Dave Rice explains in "Sustaining Consistent Video Presentation," one of several white papers commissioned by the Tate museum in London, NTSC (National Television System Committee) DV (Digital Video), converted during transcoding from a sampling rate of 4:1:1 to 4:2:0, will exhibit "substantial loss to color detail."[13] To test this hypothesis, for this paper a

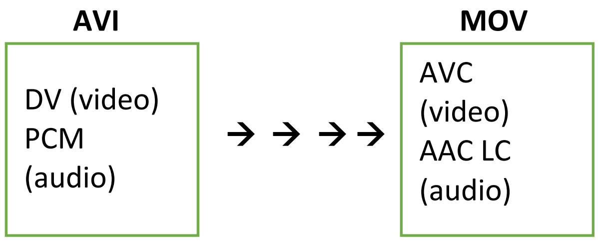
---

[11] https://premierepro.net/color-management-premiere-pro/

[12] https://www.tate.org.uk/about-us/projects/pericles/audiovisual-adherence

[13] https://www.tate.org.uk/about-us/projects/pericles/sustaining-consistent-video-presentation

34.5 MiB DV file in an AVI (Audio Video Interleave) container was downloaded and transcoded to an AVC file in a QuickTime container, as follows:

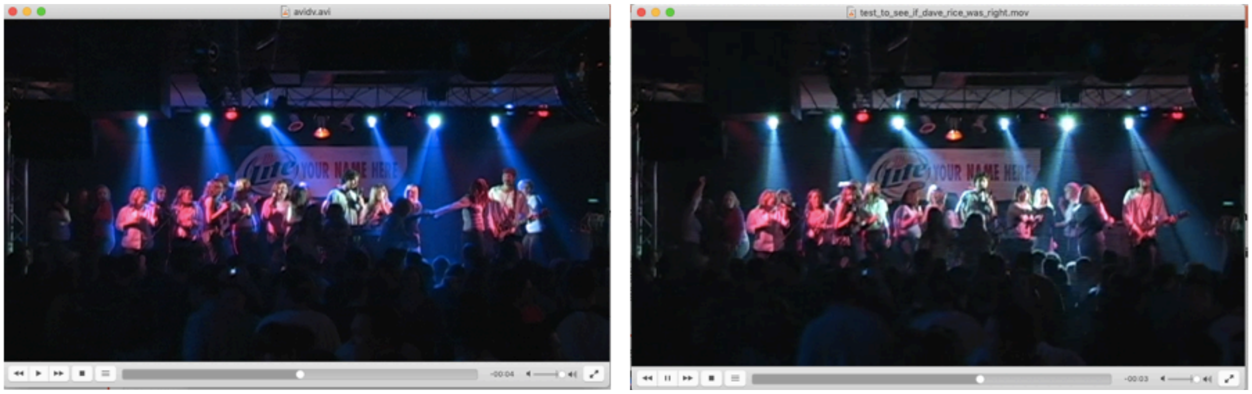


Prior to transcoding, it was assumed, of course, that the original DV file would use a sampling rate of 4:1:1, working under the assumption that “NTSC DV uses a 4:1:1 color subsampling pattern that samples color horizontally but not vertically.”[14] However, a look at the MediaInfo report described the chroma subsampling pattern as 4:2:2. At first this appeared to be one of the above-mentioned description errors. But some further research into DVDPRO50— a proprietary format developed by Panasonic in 1997 for professional use in electronic news gathering and filmmaking— revealed that this particular DV file was in some regards unique. The DVDPRO50 camera used “two codecs working in parallel” and, also unique to DV, employed a 4:2:2 chroma subsampling for improved resolution— in other words, the MediaInfo analysis was correct.[15]

However, the MediaInfo readout failed to identify the ITU-R recommended standard at use by the DV file— presumably BT.601.[16] This may explain why color shift occurred during this transcode anyway, regardless of the chroma subsampling rate:

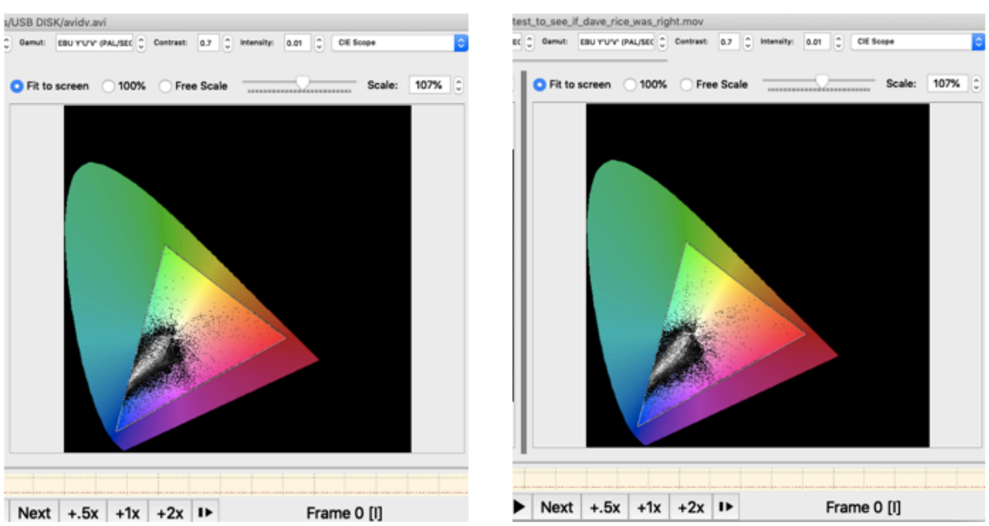
---

[14] Ibid.
[15] https://en.wikipedia.org/wiki/DV
[16] Ibid.



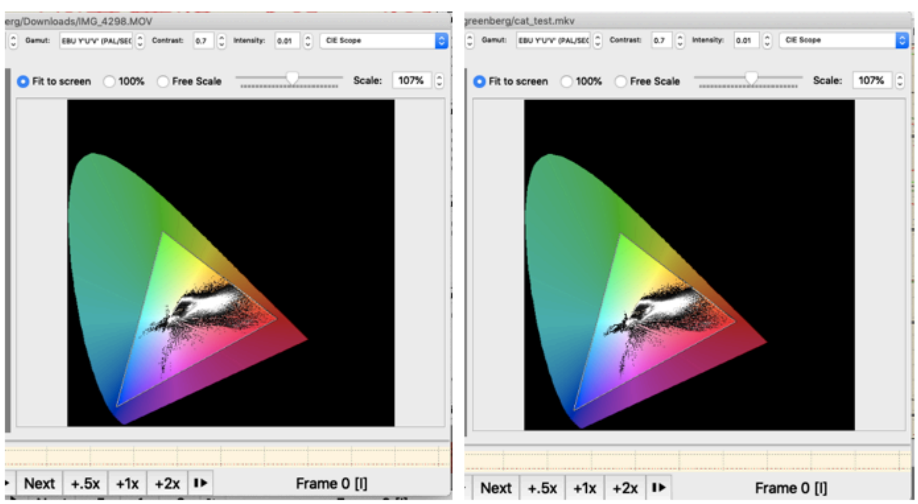
NTSC DV (.avi) → AVC codec in MOV container exhibits blue-to-green color shift

It is, however, possible (although not guaranteed) that this color shift could be corrected in ffmpeg by using the "Convert colorspace of video" script to convert the BT.601 to BT.709— or, alternately, by ensuring that the transcoding process doesn't automatically default the standard to BT.709. Either way, due to inadequate self-description in the original DV file (my identifying it as BT.601 is contextual, not empirical), no IUT-R recommendation is identified even in the MediaInfo report. Oddly, even a look at QCTool's CIE chart filter failed to display any significant differences between the two files:



Second Test: Rewrapping 4K AVC codec in QuickTime (MOV) container to Matroska (MKV)

For this test, a 1.73 MiB 4K video file was captured on an iPhone 8 and rewrapped to a Matroska (MKV) container, testing again for color fixity and self-description. This time, the ITU-R recommendation *did* appear in MediaInfo's table, clearly described as BT.709 under "Color primaries," "Transfer characteristics," and "Matrix coefficients." However, this was somewhat to be expected, as even in 4K, the iPhone 8 cannot capture uncompressed raw video data.[17] (A truly uncompressed video file would, per Dave Rice as mentioned above, rely on its eventual container for description.) The BT.709 description was accounted for in both MediaInfo analyses, and no detectable color shift occurred:



[17] https://www.shutterstock.com/blog/how-to-shoot-a-film-iphone11. As of iPhone X, however, an iPhone can capture raw still image data— but still not so for video.

Third Test: AVC in MXF (Material Exchange Format) transcoded to FFV1 in an MKV container

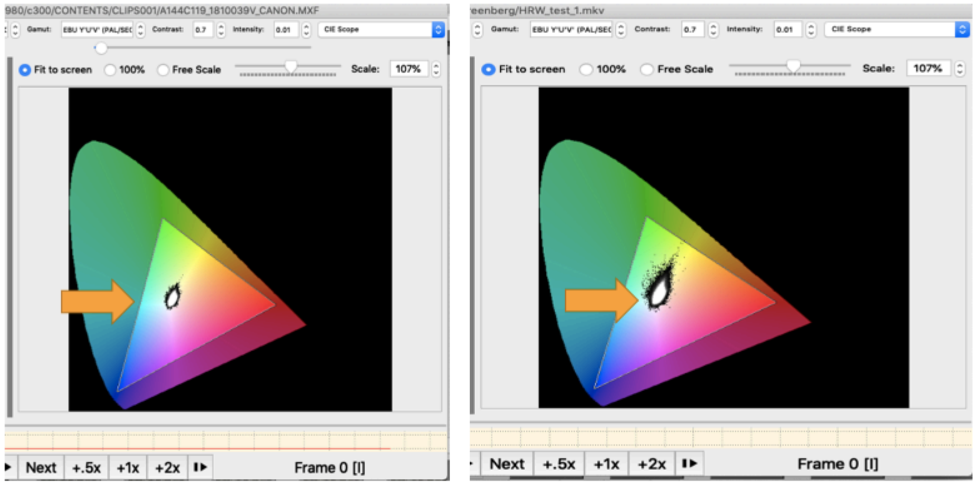
For this test, a 2.52-GiB video file MXF video file containing an AVC (H.264) codec was rewrapped to the lossless intra-frame video codec FFV1 in a Matroska (MKV) container. Likely due to the fact that this data had originally been captured raw, the ITU-R color space description was not accounted for in the MediaInfo analysis. The mystery is compounded by the fact (described in MediaInfo) that the data was captured on a Canon EOX C300 Mark II camera, which allows users the option of four distinct color spaces: BT.709, DCI-P3 (for theatrical projection), BT.2020, and a proprietary color space, Cinema Gamut, which "captures more colors than any current display technology can replicate" and is thereby "future-proofed."[18] It can be surmised that the data in question uses BT.2020, as this is the generally recommended color space for data captured in 4K; however, this cannot be assumed, since BT.709 remains the standard for web-based content.[19]

In any event, while the transcode failed to result in and discernable color shift on a 2017 MacBook Air monitor display environment. However, a discrepancy did appear between the respective achromatic regions of the CIE diagrams for each file, for reasons that remain unclear:

---

[18] https://www.usa.canon.com/internet/portal/us/home/learn/education/topics/article/2018/july/eos-c300-mark-ii-looking-forward-wide-color-spaces/eos-c300-mark-ii-looking-forward-wide-color-spaces/!ut/p/z0/nZDBbsIwEER_hR5ytHZLUUWPNBGCighxS3yJto4JS9x1cExT_r4hH1CpHGc1bzSzoKEALfTNDUX2Qm7UpX6t8sNmudmmuMNs94aH_f49W6_T7WK1gA_QfxvGhHnI07wB3VE8KZajh2KOz0sozld3g8L6XpkXRPVFoVXMynnfsjTq6MNAoVYD11YZ73xQfUfG9o8xUPL5ctEr0MZLtD8RCkPipRI7DPazmq4SE6y54UiucpaCjKEJ2vpqppfMSOoZS99xmHSCFCIbZ_sE75sSvG8agX_3e4SBrtVldhuefgHWLqcQ/ An archivist might reasonably question just how "future-proof" any proprietary standard can be!

[19] Ibid.



Applying the 2014 FADGI criteria to MOV, MXF, MKV, and AVI

Part One of the 2014 Federal Agency Digitization Guidelines Initiative (FADGI) *Digital File Formats for Videotape Reformatting* white paper lays out its *Detailed Matrix for Wrappers*, a useful tool for considering color fixity (and much else) in digital preservation workflows. One of the *Sustainability Factors* against which the wrappers are evaluated is, unsurprisingly, "Self-Documentation." Each wrapper receives a rating of *good, acceptable,* or *poor* against each criterion. Each of the three target wrappers tested for this paper— MOV, MXF, and MKV—receive a rating of *good*, as they "include a significant amount of technical metadata" and "optional descriptive metadata is also well-supported." Under a different (but related) criterion, "Native Embedded Metadata Capabilities," all three again receive a rating of *good.* AVI, also looked at as a source container for this paper, receives a rating of *acceptable* for "Self-description" and for "Native Embedded Metadata Capabilities," since, as an older format, it lacks "some modern video features" like scan type and pixel aspect ratio, which are not included as mandatory by AVI.[20]

Monitor Environment

Although touched on above, the way a digital file's color space may or may not interact or interface with a display monitor environment is worth exploring in greater depth. This is a particular problem as the usage of BT.2020 (and its very architecturally similar successor, BT.2100) gains wider acceptance. A BT.2020-native image presented on a standard BT.709 display will appear desaturated, while a BT.709-native image presented on an HD BT.2020 display will appear oversaturated.[21] At least one engineer at the Society of Motion Picture and Television Engineers (SMPTE) has considered *gamut mapping* as a potential corrective. In gamut mapping, a receiving device's gamut range is modified to accommodate media from a source device.[22] However, the best way to apply a gamut mapping workflow video files remains unclear, "given the limited processing power at the reception side."[23]

Other Factors Affecting Interoperability

Best practices dictate that an archivist managing a digital preservation workflow should start by looking at the self-descriptiveness of the source codec, and then looking at how that codec interacts with a range of containers. The most dramatic consequences of this decision frequently involve color management. An MXF wrapper— unlike MOV or JPEG2000— doesn't

---

[20] http://www.digitizationguidelines.gov/guidelines/FADGI_VideoReFormatCompare_pt1_20141202.pdf

[21] Maciej Pędzisz, "Beyond BT.709."

[22] https://graphics.stanford.edu/courses/cs178/applets/gamutmapping.html

[23] Maciej Pędzisz, "Beyond BT.709."

use a Color Parameter (‘colr’) atom to store color space metadata regarding transfer between RGB and Y’CbCr color spaces.[24] Accordingly, when the JPEG2000 codec is contained within an MXF wrapper, the decoder will read a file with a Y’CbCr color space as RGB.[25]

Frame size can also interact with color space in undesirable and frustrating ways. Scaling algorithms for frame size function in large part by “[setting] the newly introduced pixels…to values that average the luminance and color of [their] neighbors.”[26] These new pixels, in turn, introduce new colors as artefacts. If the problem isn’t caught during the QC process, these artefacts will, of course, become “baked into” the new, “normalized” transcoded files. Similarly, transcoding to a fixed frame rate can result in color shift if the source codec is insufficiently self-descriptive.[27] Here again, however, ffmpeg can easily be used to target the colorspace via the `colormatrix=src:dst` command.[28]

Conclusion

One archivist who assisted in the research conducted for this paper expressed the need to uphold access file fixity while at the same time avoiding perfectionism, noting that “compression will always be noticeable when shown side by side with the master.”[29] Digital preservation in practice is rarely black-and-white (pun intended), and a constant array of moving parts and budgeting realities proscribes just what an archivist can and can’t do.

---

[24] https://www.nyu.edu/tisch/preservation/program/student_work/2016fall/16f_1807_Campbell_a2.pdf

[25] Ibid.

[26] https://www.tate.org.uk/about-us/projects/pericles/sustaining-consistent-video-presentation

[27] https://kdenlive.org/en/project/color-hell-ffmpeg-transcoding-and-preserving-bt-601/

[28] https://amiaopensource.github.io/ffmprovisr/#convert-colorspace

[29] Email from Kelly Haydon.

Ultimately, in the absence of sufficient self-description within codecs and containers whose creators don't necessarily work with archival best practices in mind, it is up to archivists to ensure that vendors and collecting institutions work together to increase the granularity of metadata documentation— for color information and otherwise—at every step of the way, as obsolescence marches on.

Works Cited

Campbell, Savannah. "The Great Video Codec Debate: JPEG2000 and FFV1." December 16, 2016. Accessed Dec. 2019. <https://www.nyu.edu/tisch/preservation/program/student_work/2016fall/16f_1807_Campbell_a2.pdf>

"HDR FAQ." arri.com. Web. Dec. 2019. <https://www.arri.com/en/learn-help/learn-help-camera-system/camera-workflow/image-science/frequently-asked-questions-on-hdr#accordion-44200>

Dowlatshahi, Shahed. "High Dynamic Range Video: Issues Related to Moving Image Preservation." Web. Dec. 2019 <https://www.nyu.edu/tisch/preservation/program/student_work/2017fall/17f_1807_Dowlatshahi_a2_paper.pdf>

Miller, G. Tyler. *Environmental Science.* 1993: Web. Accessed Dec. 2019 via Internet Archive. <https://archive.org/details/environmentalsci00mill_0>

Oran, Andrew and Vince Roth. "Color Space Basics." *AMIA Tech Review.* May 2012, No.4. Web. Dec. 2019. <https://amianet.org/wp-content/uploads/Publication-AMIA-Tech-Review-V4-2012.pdf>

Pedzisz, Maciej. "Beyond BT.709." *SMPTE Motion Imaging Journal.* November/December 2014. Web. Accessed Dec. 2019 via JSTOR.

"ffmpeg Documentation." ffmpeg.org. Web. Accessed Dec. 2019. <https://ffmpeg.org/ffmpeg.html>

Rice, Dave. "Audiovisual Adherence." tate.org/uk. Web. Accessed Dec. 2019.

Rice, Dave. "Sustaining Consistent Video Presentation." tate.org/uk. Web. Accessed Dec. 2019. <https://www.tate.org.uk/about-us/projects/pericles/sustaining-consistent-video-presentation>

"REC.2020 COLOR SPACE." *Adobe Support Community.* <https://community.adobe.com/t5/premiere-pro/rec-2020-color-space/td-p/9157515>

Leirpoll, Jarle. "How Display Color Management in Premiere Pro Works." premierepro.net. Web. January 30, 2019. Accessed Dec. 2019. <https://premierepro.net/color-management-premiere-pro/>

Levoy, Marc. "Gamut mapping." stanford.edu. February 9, 2012. Accessed Dec. 2019. https://graphics.stanford.edu/courses/cs178/applets/gamutmapping.html

“DV.” wikipedia.org. Accessed Dec. 2019. <https://en.wikipedia.org/wiki/DV>

Simons, Loren. “EOS C300 Mark II Looking Forward: Wide Color Spaces.” usa.canon.com. April 27, 2016. Accessed Dec. 2019. <https://www.usa.canon.com/internet/portal/us/home/learn/education/topics/article/2018/july/eos-c300-mark-ii-looking-forward-wide-color-spaces/eos-c300-mark-ii-looking-forward-wide-color-spaces/!ut/p/z0/nZDBbsIwEER_hR5ytHZLUUWPNBGCighxS3yJto4JS9x1cExT_r4hH1CpHGc1bzSzoKEALfTNDUX2Qm7UpX6t8sNmudmmuMNs94aH_f49W6_T7WK1gA_QfxvGhHnI07wB3VE8KZajh2KOz0sozld3g8L6XpkXRPVFoVXMynnfsjTq6MNAoVYD11YZ73xQfUfG9o8xUPL5ctEr0MZLtD8RCkPipRI7DPazmq4SE6y54UiucpaCjKEJ2vpqppfMSOoZS99xmHSCFCIbZ_sE75sSvG8agX_3e4SBrtVldhuefgHWLqcQ/>

Wu, Rubidium. “Can You Successfully Shoot a Film on the New iPhone 11 Pro?” shutterstock.com. Accessed Dec. 2019. <https://www.shutterstock.com/blog/how-to-shoot-a-film-iphone11>

“Digital File Formats for Videotape Reformatting.” *Federal Agencies Digitization Guidelines Initiative.* December 2, 2014. Accessed Dec. 2019. <http://www.digitizationguidelines.gov/guidelines/FADGI_VideoReFormatCompare_pt1_20141202.pdf>

“Color Hell: Ffmpeg transcoding and preserving BT.601.” kdenlive.org. Accessed Dec. 2019. <https://kdenlive.org/en/project/color-hell-ffmpeg-transcoding-and-preserving-bt-601/>

“Transcode Video to a Different Colorspace.” amiaopensource.github.ffmprovisr. <https://amiaopensource.github.io/ffmprovisr/#convert-colorspace>