ACES Gamut Mapping Architecture VWG Technical Documentation Deliverable

February 1, 2021
Authors: Carol Payne, Matthias Scharfenberg, Nick Shaw

Introduction

This document is intended to detail the specification of the final CTL deliverable to be added to the ACES repository for release in ACES version 1.3. The original scope of the Gamut Mapping Architecture group was broad, so this document will include a summary of the research done in order to arrive at this deliverable, as well as workflow requirements and implementation considerations for the future Implementation VWG to consider.

History / Research

When the proposal for this group was written, the main issue was outlined as:

> Users of ACES are experiencing problems with out of gamut colors and the resulting artifacts (loss of texture, intensification of color fringes). This issue occurs at two stages in the pipeline.
>
> i. Conversion from camera raw RGB or from the manufacturer's encoding space into ACES AP0
> ii. Conversion from ACES AP0 into the working color space ACES AP1

It was acknowledged early on in the group that this artifacting can also occur in VFX/Color grading, as well as the Output Transform stages in the pipeline.

The working group chairs set the scope:

Propose transforms between color spaces that avoid or reduce color clipping. Solutions for this may include:

- Proposing a suitable color encoding space for digital motion-picture cameras.
- Proposing a suitable working color space.
- Propose a suitable gamut mapping/compression algorithm that performs well with wide gamut, high dynamic range, scene referred content that is robust and invertible.

The group started out investigating the working and encoding spaces (ACES 2065-1 and ACEScg). However, it was agreed early on that although the possibility of creating a new ACES working space which mitigates common gamut issues should not be discounted, it would require a very strong case as to the benefits. Changing a core component of ACES would potentially introduce backwards compatibility issues, and would also be based only on the situation at the current time. It raised the possibility of having to change the working space repeatedly in future. Thus, the focus moved on to the third option - a suitable algorithm to solve the artifacting while maintaining as much of the current ACES standards and structure as possible. The gamut mapping approach chosen is one of compression, and will be referred to as such from here on out. It deals with the ACES image data “as is”, and simply strives to convert that into less problematic image data.

Based on the history above, the general working assumptions were:

- Samples are relative scene exposure values (i.e. scene-referred linear data) with no assumed min/max value range boundaries.
- The gamut mapping operator is per-pixel only (i.e. not spatial or temporal)

The stated ideals for a gamut compression algorithm were:

- Exposure invariance - $f(a \cdot RGB) = a \cdot f(RGB)$
- Source gamut agnosticism
- Monotonicity
- Simplicity - suited to a fast shader implementation
- Invertibility (see caveats later in document)
- Colors in a “zone of trust” will be left unaltered

While a suitable algorithm should be able to map arbitrary gamut A into arbitrary gamut B, it should not be a requirement that all source data must be contained within gamut A. Nor is it necessarily a requirement that the output should be entirely bounded by gamut B. Indeed, allowing extreme source values to be mapped to output values close to, but not within, the target gamut means that the compression function does not need to tend to the horizontal at the boundary. This means that its inverse will not tend to the vertical, which is beneficial for invertibility.

Because the unreal colors which occur are a result of the mismatch between a camera and human observer (among other causes) and are outliers in the residual error of a transform optimized for a subset of important “memory” colors, what they “should” look like is somewhat undefined. The important thing is to remap them into values which are plausible rather than “accurate”.

What was determined to be outside the scope:

- Colorimetric accuracy or spectral plausibility of input device transforms (IDTs)
- Display gamut mapping. (Required modifications to the RRT/ODT will need to be addressed by a subsequent group.)

- Customizing for specific input/output gamuts
- Working in bounded or volume-based gamuts
- Actions which could limit creative choices further down the line (e.g. excessive desaturation)

User Testing

Once the working group settled on the baseline algorithm and its properties (discussed in the technical specification below) a set of targeted, small scale user tests were conducted to ensure the foundations of the work were solid. The testing was composed of two groups - VFX compositors and colorists. Between these two disciplines every major use case for the gamut compression algorithm could be tested and measured. The group gathered an open repository of test images that clearly exhibited the problem to be solved. It then derived a set of test scenarios for each group ranging from keying, blur, grain matching, hue adjustment, and more. The tests were conducted in Nuke and Resolve, on both SDR and HDR monitors.

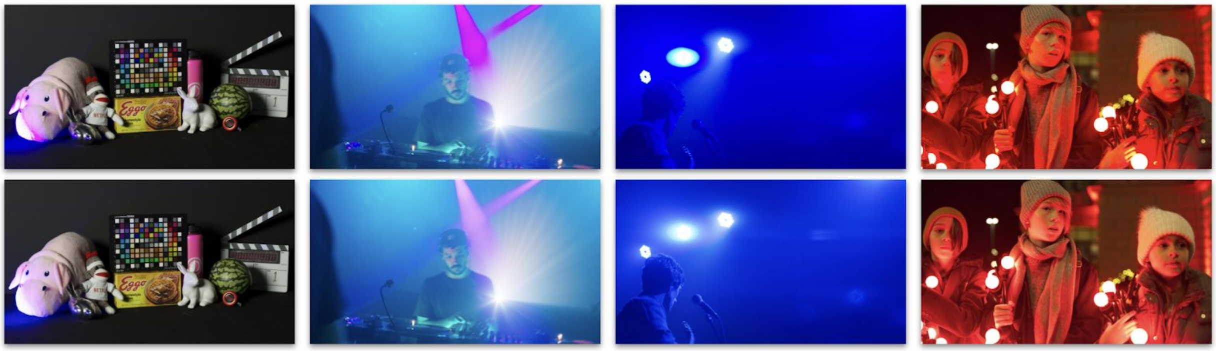


Figure 1: User Testing Footage Examples

Overall, the results of the user testing were positive and uncovered no major issues in the algorithm functionality. 75% of compositors and 96% of colorists stated that using the algorithm helped them complete their work and achieve their creative goals. For full user testing results, please refer to the working group historical repository.

Technical Specification

ACEScg values within the AP1 gamut are positive. Values outside of the AP1 gamut are negative in one or two of their components. These negative components are the problematic values to be fixed by mapping them back inside the bounds of the gamut.

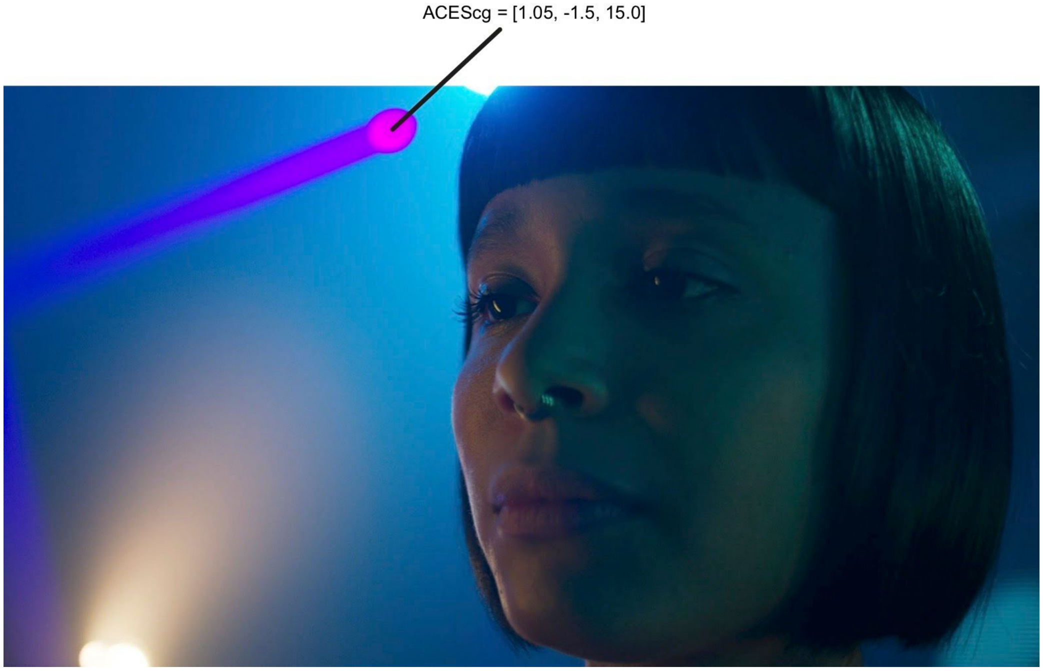

ACEScg = [1.05, -1.5, 15.0]

Figure 2: Before Gamut Compression (ACES Rec.709 Output Transform)

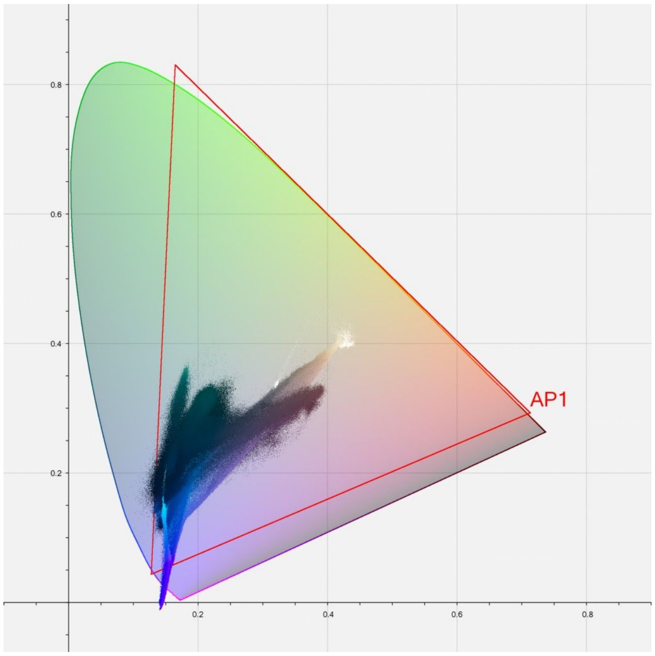


Figure 3: Before Gamut Compression (Chromaticity Plot)

The gamut compression algorithm runs per-pixel, and is controlled with a small set of parameters. These parameters control how much of the core gamut to protect, how much to compress each of the color components, and the shape of the compression curve. The algorithm operates in a purely RGB scene linear space (ACEScg).

To achieve this gamut compression, a few different representations of the input image need to be constructed:

Achromatic

- Similar to luminance, this represents the achromatic or neutral axis of the color cube. Achromatic is constructed by taking the maximum of the R, G, and B color components.

$$achromatic = max(r, g, b)$$

Construction of the neutral axis leaves one of the color components unaffected through the gamut compression transformation, which makes inversion possible.

Distance

- Similar in concept to saturation, it represents the distance of a color component from the achromatic axis.

$$distance = (achromatic - rgb)/achromatic$$

This calculation produces the distance, or the "inverse rgb ratios" normalized by dividing by the achromatic. This gives a distance of 1.0 at the gamut boundary (where one component is zero.) *See Figure 4*.

Next, distance values above 1 need to be compressed to live within the gamut. A compression function which is linear (y=x) below a specified threshold is used in the algorithm. Above the threshold, y values are mapped lower in a smooth curve. The threshold must be less than 1 in order to make space for the higher values to be compressed into gamut, but not so low as to negatively impact memory colors (see Compression Threshold). A power compression function was selected, out of many compression functions which were tested.

If infinite distance is compressed to the gamut boundary, this makes very inefficient usage of the compression space between the threshold and the gamut boundary. By setting limits, waste spent on highly improbable values is reduced and instead the algorithm is tuned towards the majority of offending pixels.

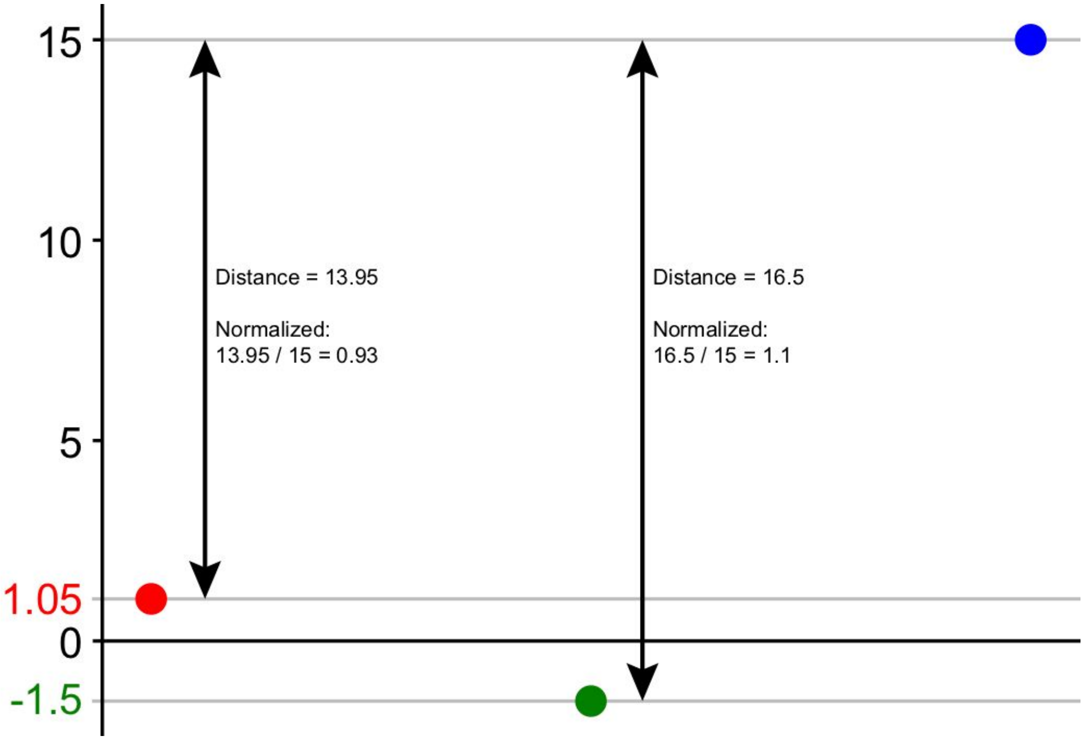


Figure 4: Normalized Distances

Parameters

Distance Limit

$[1.147, \ 1.264, \ 1.312]$

- The limits of distance from the achromatic axis that values will be compressed to the gamut boundary. The limit values have been set to constants chosen so that the encoding gamuts of all digital cinema cameras with official ACES IDTs (ARRI, RED, Canon, Sony, Panasonic) will compress to within AP1. *See Figure 5*.

Compression Threshold

$[0.815, \ 0.803, \ 0.88]$

- Percentage of the core gamut to protect. These have been set to constants derived from the boundary of the ACEScg values for the ColorChecker Classic 24 (as specified in ISO 17321-1).

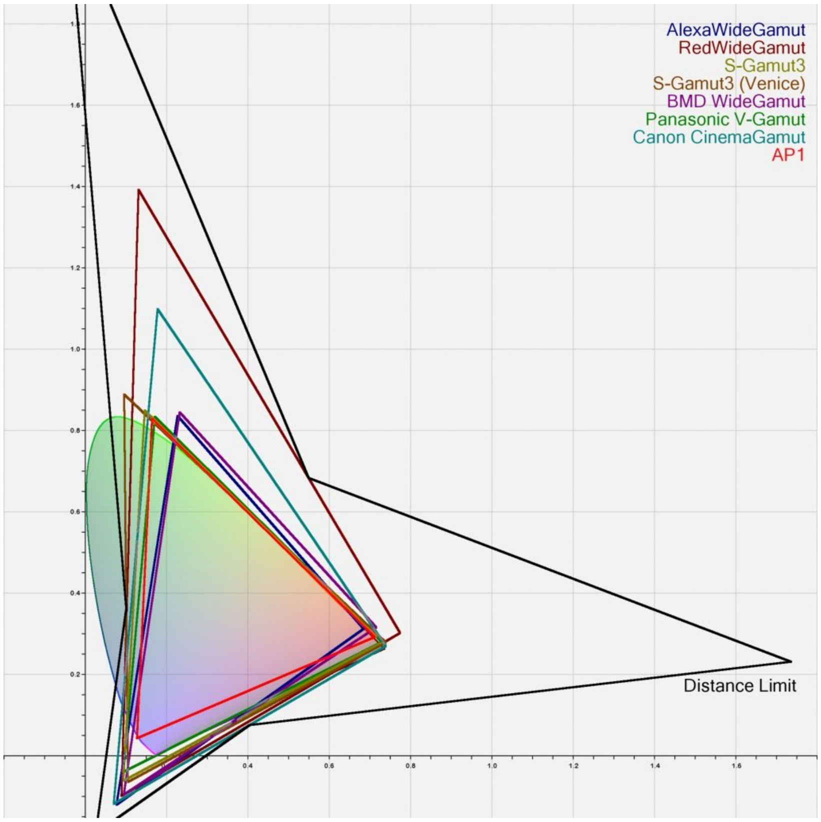


Figure 5: Distance Limit (Chromaticity Plot)

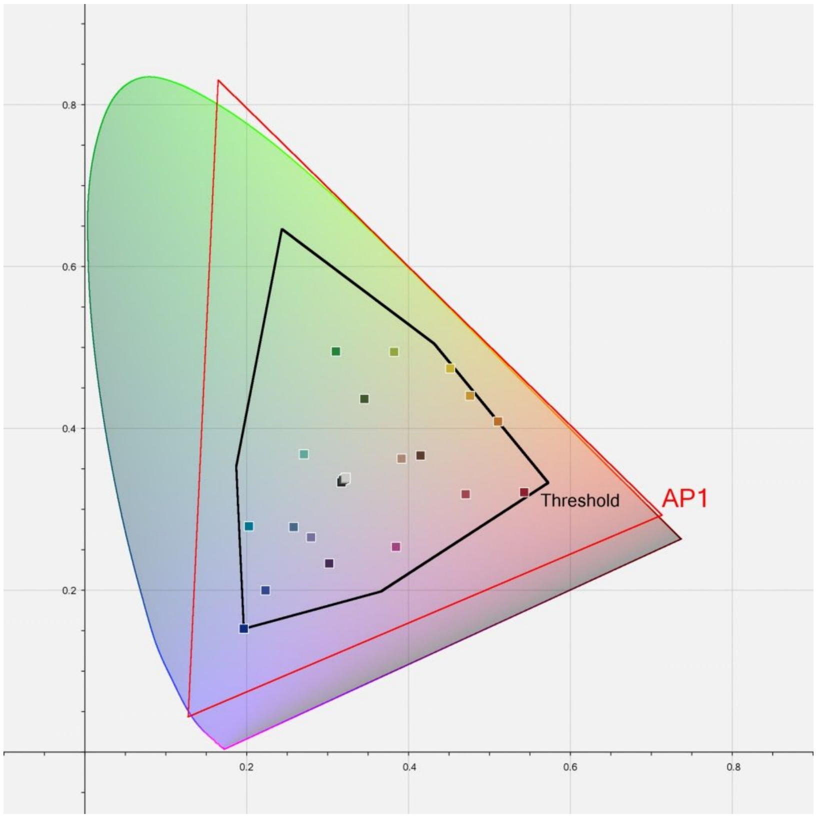


Figure 6: Compression Threshold and ColorChecker 24 Patches (Chromaticity Plot)

Power Curve Exponent

- How “aggressively” to compress values.

The value of the exponent is currently set at 1.2. There is no objective measure of the correctness of this value. It is still open to discussion whether this value should be exposed beyond the default in an implementation.

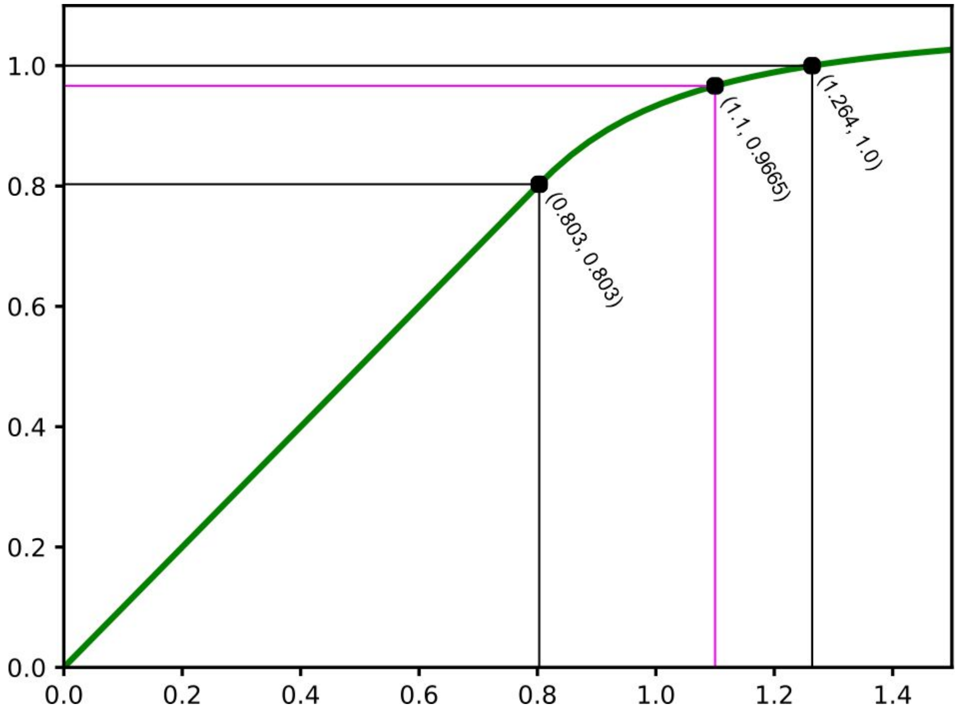


Figure 7: Green Distance Compression Curve (Threshold=0.803, Limit=1.264, Power=1.2)

Lastly, once the compressed distance has been calculated, the RGB image is reconstructed from the achromatic and newly compressed distance representations.

$$rgb \ = \ achromatic \ - \ compressed\,distance \ * \ achromatic$$

Note that the red value has also been moved slightly, because although it was not negative, its normalised distance was close to 1.0, and beyond the protected threshold for that channel (0.815).

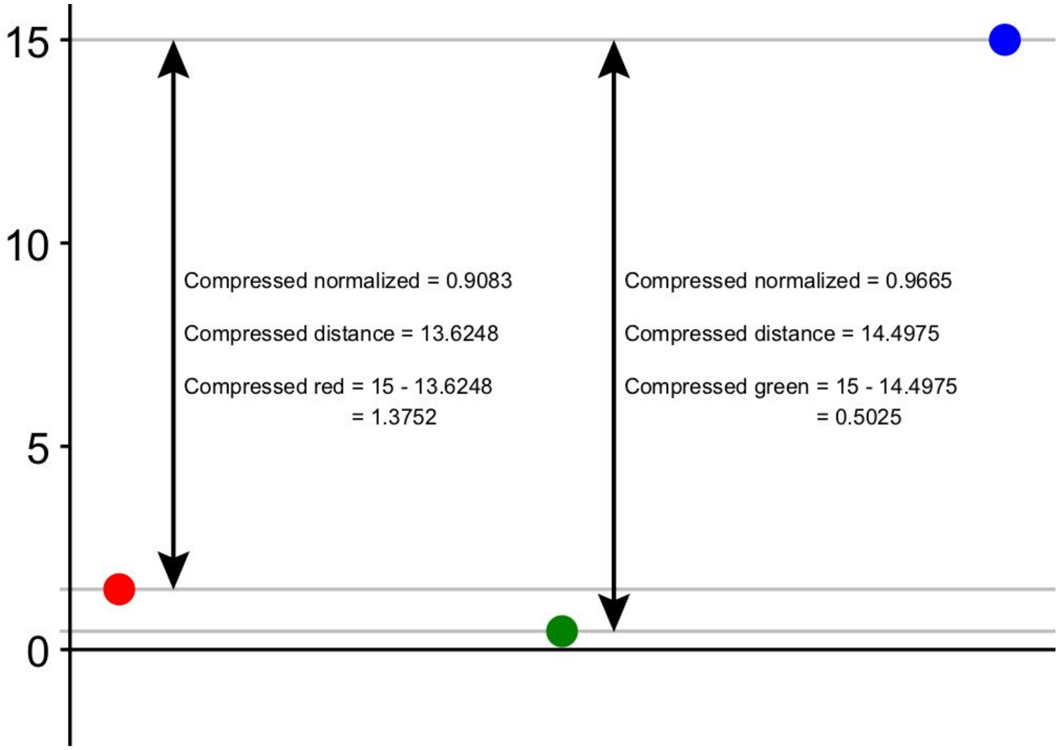


Figure 8: Compressed Distances

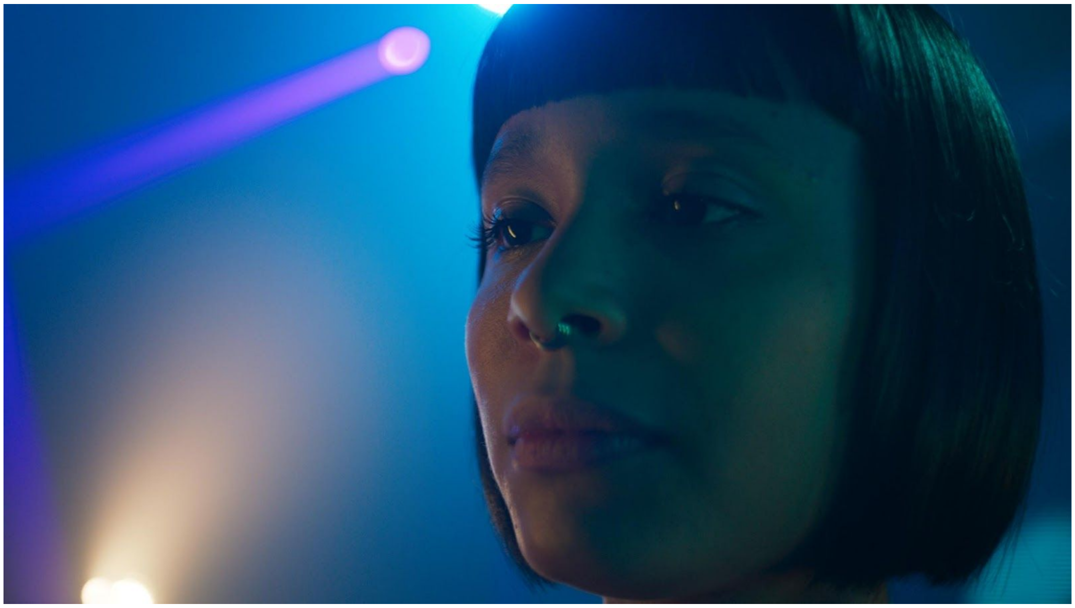
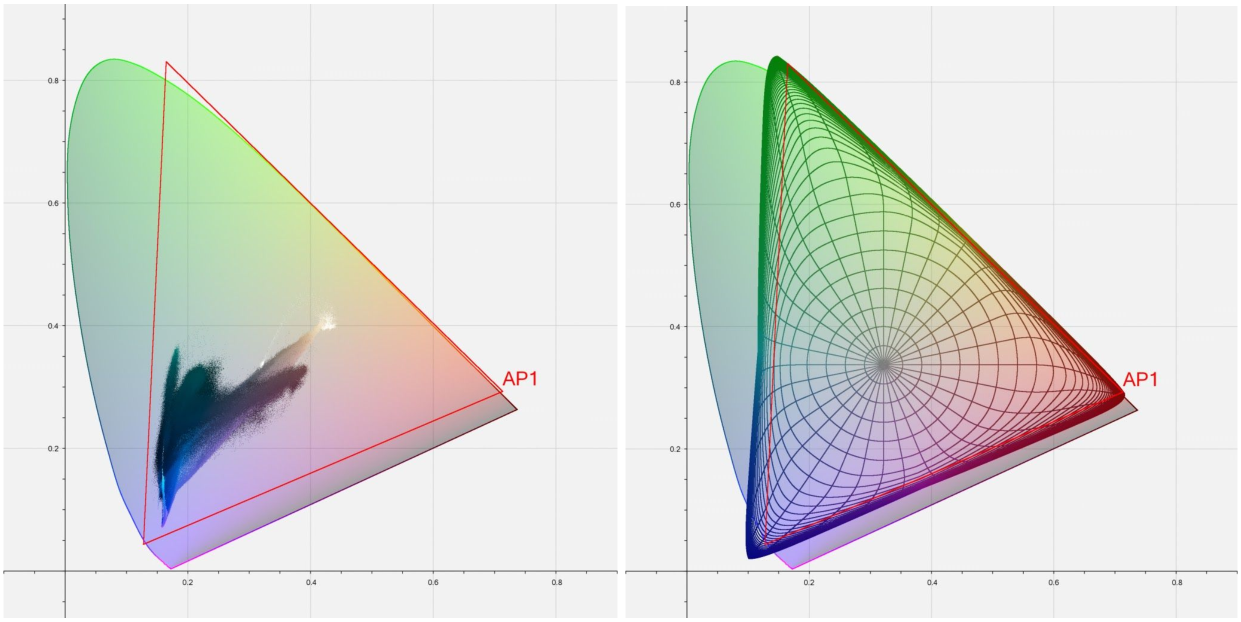


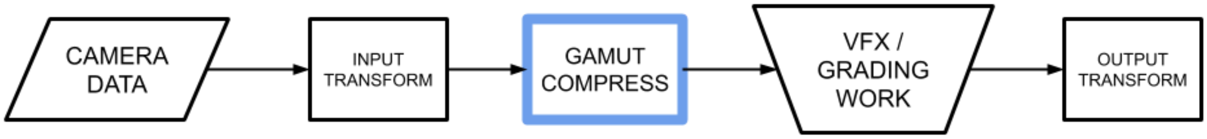
Figure 9: After Gamut Compression (ACES Rec.709 Output Transform)



Figures 10 & 11: After Gamut Compression (Chromaticity Plot) & Polar Chromaticity Grid

Workflow Requirements

One of the most important things to note is the `gamut_compress` tool is just one piece of a bigger picture. ACES is a set of blocks that piece together to form the whole imaging chain – and this is a part of one of those blocks. The `gamut_compress` algorithm operates only in a scene referred linear RGB space – in our case, ACEScg. Colorists will still grade in ACEScct.



The above process diagram outlines the intended workflow. The exact implementation will differ per tool, but the order of operations should remain the same. The gamut_compress algorithm should be applied immediately following the IDT, before any other work is done.

Please note that this algorithm was derived and tested using the current ACES 1.1 Output Transforms (RRT/ODT) - though it was tested under multiple other non-ACES rendering transforms. As the operation is performed in scene linear RGB values, it is completely independent from the rendering transform. The goals and types of gamut mapping needed for a rendering transform are wholly different in goal and end result, and the `gamut_compress` algorithm should not be used in comparison to or as a substitute for proper display gamut mapping.

The last important thing to note is that the gamut_compress algorithm is a per-pixel operation - but fast enough to run in real-time. This individual pixel evaluation enables the algorithm to only affect the necessary parts of the image - leaving the rest untouched, preserving as much of the originally captured image as possible. This tool is meant as a direct replacement for the “Blue Light LMT” and is intended as an optional part of the ACES pipeline. It is intended to be applied solely to problem footage, but could be applied to all footage if necessary with no adverse effects in the majority of cases.

Implementation Considerations

Invertibility

Invertibility of the transformation is an aspect that was discussed at length. The consensus was that while an inverse transform should be defined it comes with the caveat that gamut expansion can create undesirable results when used with highly saturated pixel values, such as those added as part of graphics or CG rendered imagery. It will be for the implementation working group, in conjunction with the software manufacturers, to decide on the best possible workflows to utilize the proposed gamut compression algorithm and whether that should include inversion. Theoretically, as this operation is considered more of a “pixel healing” technical operation, inversion should not be a required part of the workflow. It is more akin to a despill after pulling a key, or a bit of sharpening on a scale operation.

Default Values

The architecture working group has submitted a CTL file with default values as described above in the technical specification for the limit and compression thresholds. These values are empirically derived and as such, should not need modification by the end user the majority of scenarios. However, the default value of 1.2 for the power curve value is set based on the testing of the working group members and the small set of user testing. The value needs to remain above one, but cannot go much higher than 1.5. In testing it was noted that for SDR, a value of 1.1 or 1.2 seemed most appropriate, however for HDR monitors capable of highly saturated colors, a value of 1.4 maintained the intent better. Therefore, it was felt that this value might be the only one worth exposing as a parameter to the end user, especially in a color grading scenario, as it could affect the aesthetic look.

3D LUT Approximation

When implementing the gamut compression transform on legacy systems using a 3D lookup table (LUT) special consideration must be given to the shaper function. Common functions for implementing ACES transforms as 3D LUTs are the Log2 shaper or ACEScct. However neither

of those are designed to cover the negative value range required to map the negative components of out of gamut colors into a normalized 0 to 1 domain. For a successful 3D LUT implementation the normalizing shaper function must also cover a significant range of negative values. One method of achieving this is a logarithmic function with a linear segment, similar to the ACEScct function, but that is also reflected by its intersection point with the Y-axis as shown in Figure 12.
In order to reduce interpolation errors a lattice with a sufficient number of samples should be used (e.g. 65x65x65).

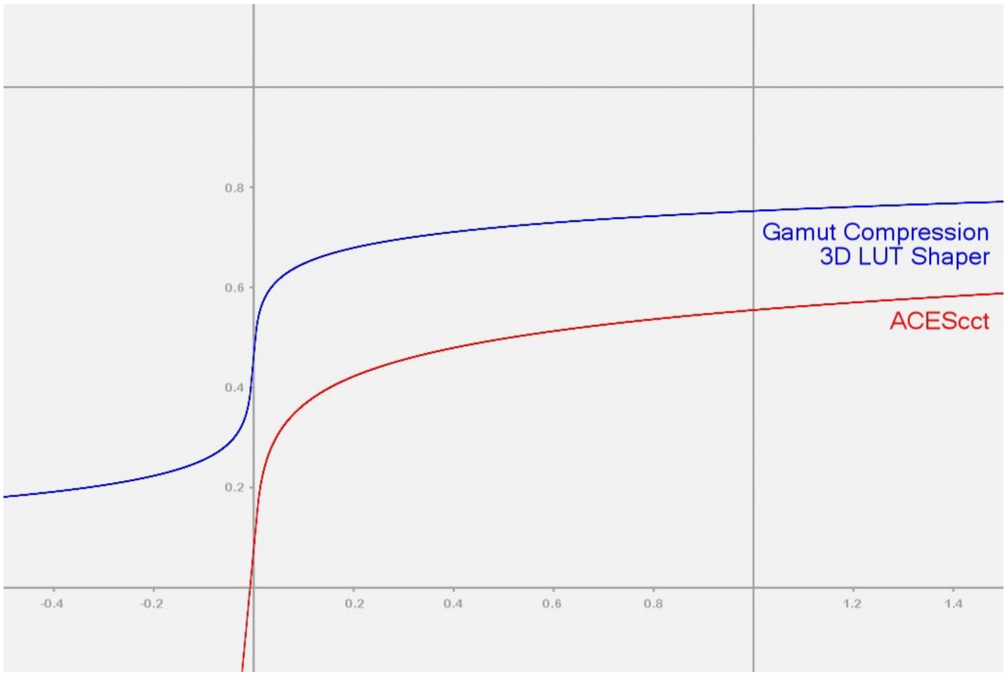


Figure 12: Example 3D LUT shaper function compared to ACEScct

Note: due to the residual error a 3D LUT approximation of the transform should be considered to be non-invertible.

Tracking

Tracking the application and possible parameters of the gamut compression algorithm would obviously be ideal. Depending on the decisions made in the implementation working group, the level of tracking necessary may vary. The ACES Metadata File is the obvious choice for the tracking vehicle, and the implementation group should work in tandem with the AMF group and ACES leadership to determine the correct course of action and priority of work.