Invertible Gamut Compression in a Perceptual Polar Space

By Nick Shaw, nick@antlerpost.com

1. Introduction

This method of gamut compression was developed for use in the ACES 2.0 Output Transforms, and presented to the Virtual Working Group, for use with a display rendering based on a modified Hellwig 2022 Colour Appearance Model (CAM). It would however be equally applicable to the compression of out of gamut values in any similar colour model.

The Hellwig model represents colour using a number of different correlates, but for the purposes of display rendering in the ACES 2.0 Output Transform, three are used:

- J : perceived lightness
- M : perceived colourfulness
- h : perceived hue

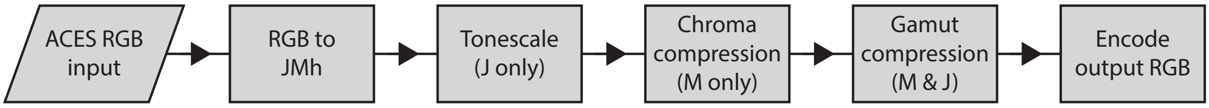

Fig 1: The flow of the ACES 2.0 Display Rendering Transform (DRT)

Gamut compression aims to compute a representation of a colour which lies outside a target gamut (the range of colours a particular display is able to reproduce) such that the new representation lies within the target gamut, while maintaining as much of the perceived appearance of the original colour as possible.

A base assumption is that when compressing a colour to within the gamut available on a given target display, the correlate of hue should be left unchanged. Therefore compression will take place in a two dimensional space, whose axes are M and J, representing a 'slice' in hue. The boundary of the corresponding slice of the target gamut has been shown to be well approximated by curved lines using a power law (gamma) function joining the maximum and minimum values on the J-axis to a cusp which represents the lines joining the primary and secondary colours in the RGB cube of the target gamut.

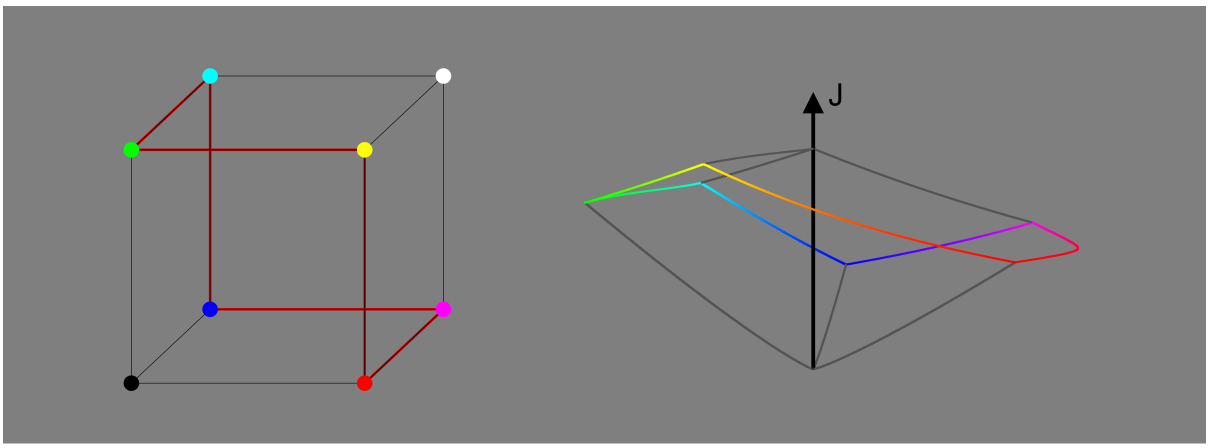

Fig 2: The cusp path shown in an RGB cube and JMh 3D polar plot

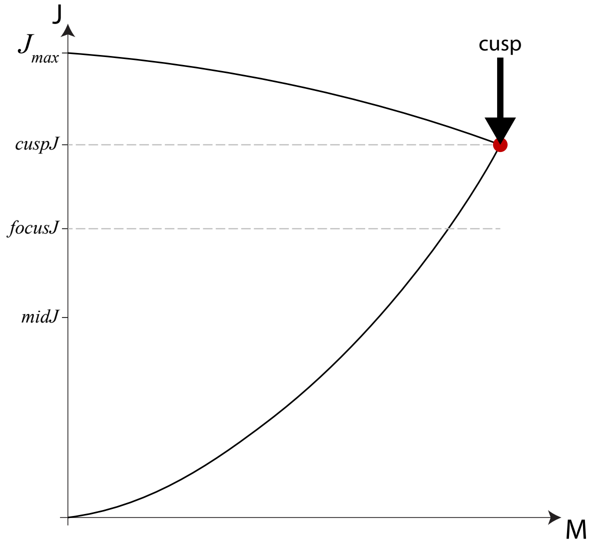

Fig 3: A 'hue slice' of a gamut plotted in M vs. J

To bring a colour within the target gamut, it must be moved towards the J-axis. In order to reach the gamut boundary at a larger M value, thus preserving more colourfulness, it has been found to be beneficial to compress along a vector which is not perpendicular to the J-axis. While intuitively it might seem that values above the cusp should be compressed along a vector which decreases J, and those above it with one which increases J, it has been found by experimentation that an aesthetically preferable result is achieved by choosing a J value ($focusJ$) half way between that of the cusp and that of mid grey, and compressing J values above that downwards, and values below it upwards.

2. The Compression Vector

The compression vector is defined by a straight line, so hs an equation with the classic form of a straight line, $y = m \cdot x + c$. In this case the y-axis represents J, and the x-axis represents M, so the equation becomes:

$$J = slope \times M + intersectJ$$

Where $slope$ is the gradient of the line and $intersectJ$ is the J value where it hits the J-axis.

For the line to have positive slope above $focusJ$, and negative slope above it, it must be horizontal (zero slope) at $focusJ$. It has also been agreed that it is desirable for the slope to be zero at J values of zero and J_{max}, where J_{max} is the J value for the peak white of the target display. The formula for slope chosen to achieve this is:

$$slope = \begin{cases} \dfrac{intersectJ \times (intersectJ - focusJ)}{slopeGain \times focusJ} & \text{if } intersectJ \leq focusJ \\ \dfrac{(J_{max} - intersectJ) \times (intersectJ - focusJ)}{slopeGain \times focusJ} & \text{if } intersectJ > focusJ \end{cases}$$

Where $slopeGain$ is a constant controlling how steep the slope becomes between the horizontals. An interactive version of the above equation can be seen in [this Desmos plot](#).

In the ACES 2.0 transform, the value of $slopeGain$ is calculated as a user parameter (called $Focus\ Distance$) multiplied by J_{max}, to scale it with peak lightness.

3. Invertibility

This equation for compression slope is controlled only by $intersectJ$ ($slopeGain$ being a user parameter and $focusJ$ being a constant for a given hue). So if the value of $intersectJ$ which produces a line passing through a particular set of (M, J) coordinates is solved for, and new coordinates (M', J') are calculated by compressing along that line, the same value will be found when solving for the $intersectJ$ which leads to a line passing through (M', J'). Thus, providing an invertible compression curve is used, an inverse compression will map (M', J') back along the same line, and result in (M, J).

Because the slope is zero when $intersectJ$ is equal to $focusJ$, when solving for intersectJ at (M, J) the value of J can be used in place of $intersectJ$ in the condition for the slope equation. Thus, substituting the $slope$ expression into the equation of the line:

When $J \leq focusJ$:

$$J = \frac{intersectJ \times M \times (intersectJ - focusJ)}{slopeGain \times focusJ} + intesectJ$$

This can be rearranged to:

$$intersectJ^2 \times \frac{M}{slopeGain \times focusJ} + intersectJ \times \left(1 - \frac{M}{slopeGain}\right) - J = 0$$

This has the form of a quadratic equation in $intersectJ$, and therefore the solution can be found using one of the variations of the quadratic formula.

Similarly when $J > focusJ$:

$$J = \frac{M \times (J_{max} - intersectJ) \times (intersectJ - focusJ)}{slopeGain \times focusJ} + intersectJ$$

This can be rearranged to:

$$intersectJ^2 \times \frac{M}{slopeGain \times focusJ} - intersectJ \times \left(1 + \frac{M}{slopeGain} + \frac{J_{max} \times M}{slopeGain \times focusJ}\right) + \frac{J_{max} \times M}{slopeGain} + J = 0$$

This also has the form of a quadratic equation in $intersectJ$, and the two can be expressed using the form of the quadratic equation required for use of a formula solution:

$$a \cdot x^2 + b \cdot x + c = 0$$

Where the values of a, b and c are:

$$a = \frac{M}{slopeGain \times focusJ}$$

$$b = \begin{cases} 1 - \frac{M}{slopeGain} & \text{if } J \leq focusJ \\ -\left(1 + \frac{M}{slopeGain} + \frac{J_{max} \times M}{slopeGain \times focusJ}\right) & \text{if } J > focusJ \end{cases}$$

$$c = \begin{cases} -M & \text{if } J \leq focusJ \\ \frac{J_{max} \times M}{slopeGain} + M & \text{if } J > focusJ \end{cases}$$

Due to division of small numbers as J and M both approach zero, the 'classic' quadratic equation formula can suffer from numerical precision issues when this solve is implemented in single precision 32-bit float. So an alternate formulation is preferable.

This is:

$$x = \frac{2 \cdot c}{-b \pm \sqrt{b^2 - 4 \cdot a \cdot c}}$$

Although quadratic equations have two roots, in this case it is not necessary to find both roots for each condition. In each case, one of the roots is always outside the zero to J_{max} range for input J and M values in the meaningful range for this use case.

Thus in this case:

$$intersectJ = \begin{cases} \frac{2 \cdot c}{-b + \sqrt{b^2 - 4 \cdot a \cdot c}} & \text{if } J \leq focusJ \\ \frac{2 \cdot c}{-b - \sqrt{b^2 - 4 \cdot a \cdot c}} & \text{if } J > focusJ \end{cases}$$

This result can then be used in the slope equation to give both constants needed for the equation of the line along which compression is to take place.

4. Compression

The compression method operates by taking the M value of the source colour, and normalising by the M value of the intersection of the compression line and target gamut boundary ($boundaryM$, see Section 5). A parametric compression curve is then applied, with the three parameters being threshold (t), limit (l) and exponent (p). The threshold is the normalised value below which the input is unchanged. The limit is the normalised value which will be compressed to the gamut boundary (a normalised value of 1.0). The exponent controls the aggressiveness of the roll-off above the threshold.

The compression curve used in the ACES 2.0 display rendering gamut compression is referred to as the ***powerP*** curve. This is the same curve as used in the ACES 1.3 Reference Gamut Compression. The equations are as follows.

First a scale factor is calculated:

$$s = \frac{(l - t)}{\left(\left(\frac{1 - t}{l - t}\right)^{-p} - 1\right)^{\frac{1}{p}}}$$

This is then used in the compression function:

$$fx) = \begin{cases} x & \text{if } x < t \\ t + \frac{x - t}{\left(1 + \left(\frac{x - t}{s}\right)^{p}\right)^{\frac{1}{p}}} & \text{if } x \geq t \end{cases}$$

After compression the compressed M value is multiplied back by the normalisation factor ($boundaryM$). The compressed J value can then be found by using the compressed M value in the compression straight line equation.

5. Boundary Intersection

Since the gamut boundary curvature is approximated by a power law (gamma) curve with non-integer exponent, and it is not possible to analytically find the intersection of such a curve with a straight line, an approximation of the intersection is needed. As the gamma curve itself is only an approximation of the true gamut boundary, provided that the approximation method for the intersection leads to points which lie on a curve similar in shape to the gamma curve, the result is as valid an approximation as the gamma curve. As the slope and offset of the compression line are defined in terms of the J-axis intersection ($intersectJ$) rather than the source J and M values, and the compressed and uncompressed values both lie on that same line, an intersection method based only on slope and offset will yield the same result in both directions, ensuring invertibility of the compression.

The intersection of two straight lines is mathematically simple to find. So a first order approximation of the intersection of the curved gamut boundary with the compression line would be the intersection of that line with the straight line joining the cusp to the origin (black) or to the peak of the target gamut (white) depending on whether the line passes above or below the cusp. This can be easily ascertained by using the cusp M value in the compression line equation, and determining whether the resulting J value is greater or less than the cusp J value.

The straight line intersection approximation will give a value which is too small, assuming the gamma curve is bending the boundary outwards. A method is needed to increase the value found in the middle part of the curve, but converging to return exactly the straight line intersection result at either end. Looking only at the lower part of the boundary (an inverted version of the same thing will work for the upper part) a good approximation of the necessary modification of the straight line intersection can be found by raising the $intersectJ$ value to the reciprocal of the exponent used by the boundary curve, and then finding the straight line intersection. This will not alter the result at the origin, but an appropriate normalisation factor is needed to divide $intersectJ$ by prior to exponentiation, and to then multiply it by afterwards, so that the result at the cusp is not affected either. This normalisation factor is found as the J-axis intersection of the compression line which passes through the cusp, by putting the J and M values of the cusp into the same quadratic intersection solve formula (see Section 3) as was used to find the J-axis intersection for the source. This value is referred to as $intersectCusp$.

So:

$$intersectJ' = intersectCusp \times \left(\frac{intersectJ}{intersectCusp}\right)^{\frac{1}{\gamma}}$$

At the intersection of a line from $intersectJ'$ with gradient $slope$ and a line from the origin to ($cuspM$, $cuspJ$):

$$intersectJ' + boundaryM \times slope = boundaryM \times \frac{cuspJ}{cuspM}$$

Solving for $boundaryM$:

$$boundaryM = \frac{intersectJ' \times cuspM}{cuspJ - slope \times cuspM}$$

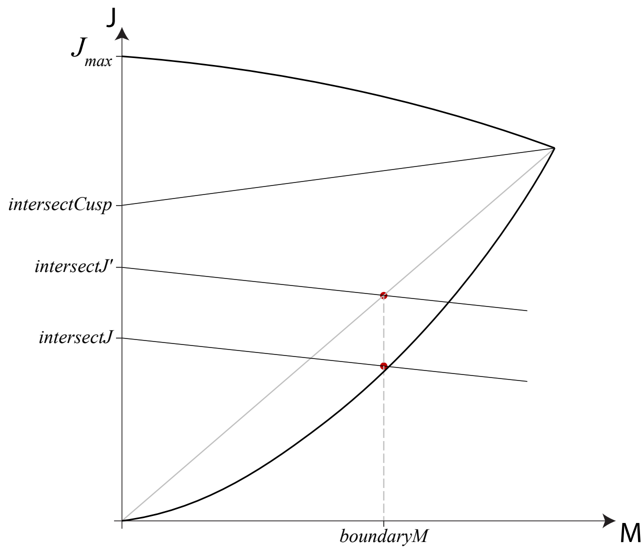

Fig 4: Approximation of intersection with a gamma curve

As can be seen from Figure 4, the value of *boundaryM* is not precisely the *M* value of the true intersection with the curve. However, as mentioned previously, because the approximation follows a path similar in shape to the gamma curve, this can be used as an approximation of the true gamut boundary. For the upper part, the same methodology can be used, simply replacing *intersectJ* with *Jmax - intersectJ*, and the same for the cusp intersection.

So if $intersectJ + slope \times cuspM > cuspJ$:

$$intersectJ' = J_{max} - (J_{max} - intersectCusp) \times \left(\frac{J_{max} - intersectJ}{J_{max} - intersectCusp} \right)^{\frac{1}{\gamma}}$$

Thus:

$$boundaryM = \frac{cuspM \times (J_{max} - intersectCusp) \times \left(\frac{J_{max} - intersectJ}{J_{max} - intersectCusp} \right)^{\frac{1}{\gamma}}}{slope \times cuspM + J_{max} - cuspJ}$$

Note: The γ values for the upper and lower parts are not necessarily the same. In the ACES 2.0 rendering, a constant value is adequate for the lower part of the gamut hull, but a varying value is needed to approximate the upper part.

This Desmos plot shows the intersection solves, slope calculation and boundary approximation, as the source (M, J) and cusp are dragged interactively.

6. 'Reach-mode' gamut compression

For the compression function described in section 4, constant values may be used for the threshold and limit. However it has been proposed that for the ACES DRT it is beneficial if the limit is set such that compression maps the boundary of a particular 'reach gamut' (ACES AP1 has been decided to be suitable) to the target gamut boundary. This ensures that when the inverse gamut compression is applied, values within the target gamut will be mapped to values inside the reach gamut. This mapping will map points on the boundary of one gamut to the boundary of the other, but will not necessarily map the primaries of one gamut to those of the other, because the mapping occurs along lines of constant perceptual hue as defined by the JMh space.

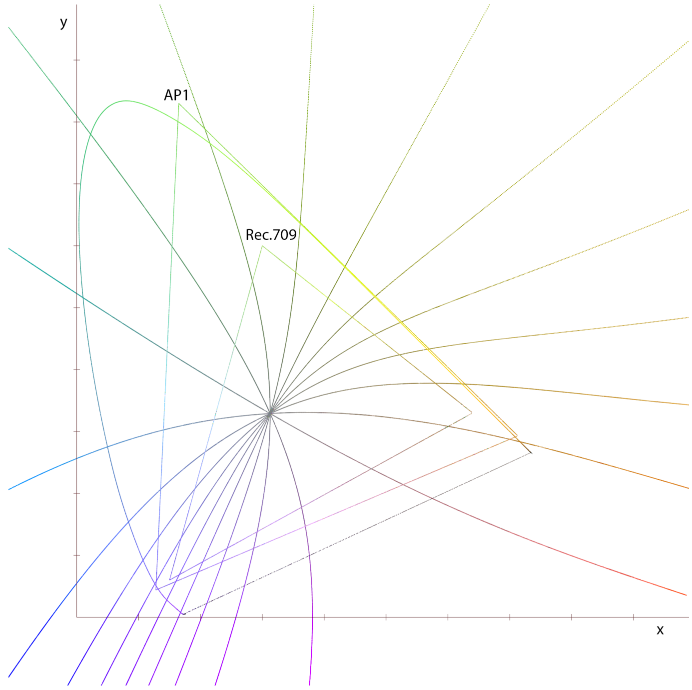

Fig 5: CIE 1931 Chromaticity plot of lines of constant JMh hue

To achieve this, another boundary intersection needs to be found – that of the compression vector with the reach gamut boundary. This boundary will have the same shape as the lower part of the target gamut but, being scene-referred, it is unbounded so the curve will continue up indefinitely, rather than tapering above a cusp, as the target gamut does.

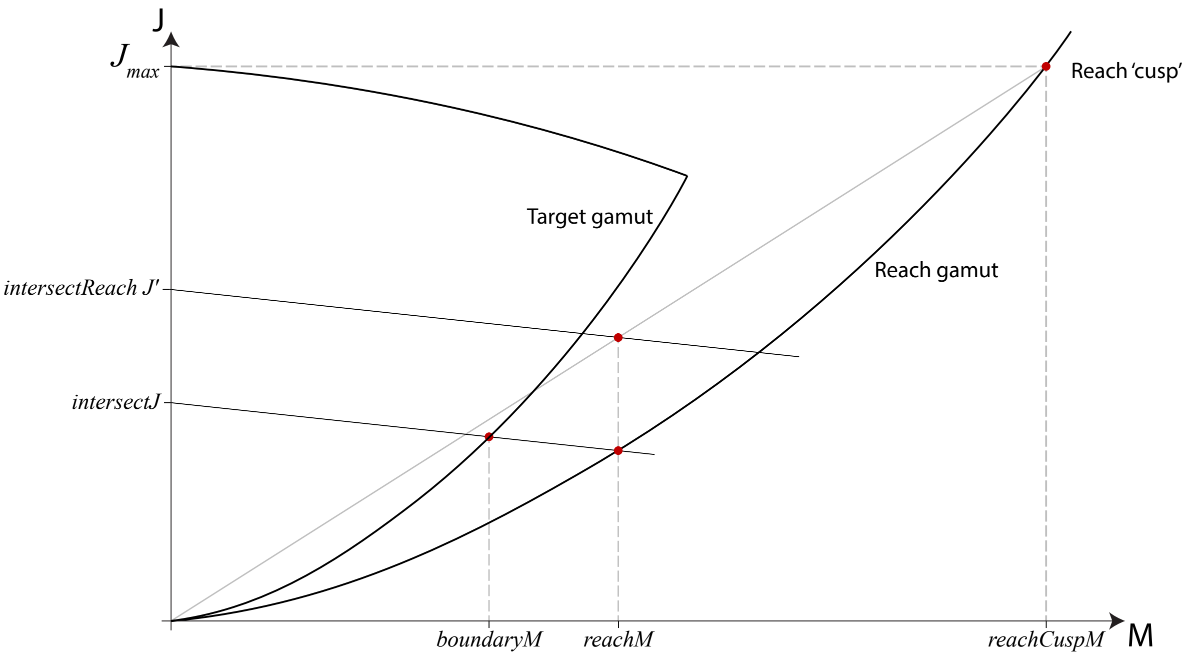

Fig 6: Intersections with the target and reach gamuts

Therefore a point on the reach gamut boundary at the hue under consideration must be found, which can then be treated in the same way as the target gamut cusp for the purposes of fitting a power law approximation to the true shape. The same γ value can be used as for the lower part of the target gamut.

If the sampled point used as a cusp for the reach gamut has a J value equal to J_{max}, then the compression vector for this point will be horizontal. Thus $Jmax$ can be used as the normalisation factor for $intersectJ$ in the same boundary intersection approximation as used for the target gamut.

Thus:

$$intersectReachJ' = J_{max} \times \left(\frac{intersectJ}{J_{max}} \right)^{\frac{1}{\gamma}}$$

and:

$$reachM = \frac{intersectReachJ' \times reachCuspM}{J_{max} - slope \times reachCuspM}$$

and the limit value for reach-mode compression is:

$$limit = \frac{reachM}{boundaryM}$$

There are no objectively correct values for the threshold and exponent parameters. An exponent of 1.2 has been found to produce satisfactory results, and is the same exponent used in the ACES 1.3 Reference Gamut Compression. For the threshold, fixed values of about 0.75 have been found to work well, but later (v049+) versions of the ACES 2.0 candidate DRT use a variable threshold, proposed by Pekka Riikonen, which is the reciprocal of the limit value. This has been shown to reduce certain artefacts for HDR display targets.