Specification

S-2013-001

ACESproxy – An Integer Log Encoding of ACES Image Data

The Academy of Motion Picture Arts and Sciences

Science and Technology Council

Academy Color Encoding System (ACES) Project Committee

July 18, 2016

Summary: This specification defines integer encoding and decoding functions for the Academy Color Encoding System (ACES). The Academy Color Encoding Specification (ACES) defines a half-precision floating-point color encoding method using a fixed set of RGB primaries. To transport and process ACES compatible images in systems that only support integer data types, this specification defines logarithmic 10-bit and 12-bit integer encodings in a smaller color space known as ACESproxy and specifies functions for conversion of this encoding to and from the floating-point encoding.

NOTICES

Revision History

Version	Date	Description
1.0	05/10/2013	Initial Version
1.1	08/02/2013	Modify ACESproxy to handle negative ACES values
2.0	12/19/2014	Modify ACESproxy primaries, constrain to legal range
2.0.1	04/24/2015	Formatting and typo fixes
2.0.2	03/11/2016	Typo fixes
	03/29/2016	Remove version number - to use modification date as UID
	07/18/2016	Change $ACESproxyLin$ to lin_{AP1}

Related Academy Documents

Document Name	Description
S-2008-001	Academy Color Encoding Specification (ACES)
S-2014-003	ACEScc – A Logarithmic Encoding of ACES Data for use within Color Grading Systems
S-2014-004	ACEScg – A Working Space for CGI Render and Compositing

Table of Contents

Introduction

This document specifies a logarithmic-type integer encoding for the Academy Color Encoding System (ACES) for use in 10-bit and 12-bit hardware systems that need to transmit a representation of an ACES floating-point image.

The Academy Color Encoding Specification prescribes a digital encoding method using the IEEE half-precision floating-point encoding defined in IEEE 754-2008. Many production systems do not support the use of 16-bit-per-color component transmission especially where hardware video systems are utilized. Some systems used for preview, look creation, and color grading are limited to common 10-bit and 12-bit video signals. In some cases, it is still necessary for a user to see a representation of the ACES image without it having been rendered for the output device using the Reference Rendering Transform (RRT), and where no 16-bit floating-point capability exists in the hardware and software.

This document specifies encoding of ACES using 10-bit or 12-bit integer data types for compatibility with those systems. This encoding is defined and named herein as ACESproxy.

10-bit and 12-bit integer data types cannot store the full range of ACES data with the same level of precision provided by the ACES half-precision floating-point format. To make appropriate use of the limited range of the 10-bit and 12-bit integer data types, the ACESproxy encoding uses a middle portion of the possible range of ACES values and is encoded using a logarithmic transfer function. To better facilitate on-set look creation, ACESproxy also uses a smaller color gamut. ACES values outside of this encoded range cannot be transmitted using ACESproxy and are assumed to be clipped to the maximum and minimum ACESproxy values.

ACESproxy is appropriate as a working-space encoding for on-set preview and look creation systems since those systems are typically designed to work with other image data encoded in a similar fashion. The ACESproxy encoding is specifically designed to work well when graded using the American Society of Cinematographers Color Decision List (ASC CDL).

The ACESproxy encoding was designed for the transmission of images across transports such as High Definition Serial Digital Interface (HD-SDI), for use within hardware systems limited to 10 or 12-bit operation, and for the creation of look metadata such as ASC CDL values.

ACESproxy images are designed to be viewed through an ACES viewing pipeline as detailed in Appendix A. When viewed without an ACES output transform, ACESproxy images are dim, low in contrast and saturation, but allow all of the sensors image data to be viewed.

ACESproxy-encoded images are intermediate encodings and are not replacements for ACES image data in postproduction color grading or finishing environments. There is no image file container format specified for use with ACESproxy. ACESproxy encoding is specifically not intended for interchange, mastering finals, or archiving, all of which are better completed using the original ACES files.

1 Scope

This document specifies 10-bit and 12-bit integer encodings of ACES for use with imaging systems that produce look metadata such as ASC CDL, and with transport systems such as HD-SDI. The color encoding provided in this format represents ACES relative exposure values as RGB triplets in a logarithmic encoding, and does not define the interfaces or signals that may carry this encoding.

2 References

The following standards, specifications, articles, presentations, and texts are referenced in this text:

SMPTE ST 2065-1:2012, Academy Color Encoding Specification (ACES)

SMPTE RP 177:1993, Derivation of Basic Television Color Equations

3 Terms and Definitions

The following terms and definitions are used in this document.

3.1 Academy Color Encoding Specification (ACES)

RGB color encoding for exchange of image data that have not been color rendered, between and throughout production and postproduction, within the Academy Color Encoding System. ACES is specified in SMPTE ST 2065-1.

3.2 American Society of Cinematographers Color Decision List (ASC CDL)

A set of file formats for the exchange of basic primary color grading information between equipment and software from different manufacturers. ASC CDL provides for Slope, Offset and Power operations applied to each of the red, green and blue channels and for an overall Saturation operation affecting all three.

3.3 High Definition Serial Digital Interface (HD-SDI)

Transport specifications for a set of bit-serial data structures for digital component signals. HD-SDI allowing 10-bit and 12-bit 4:4:4 signals is specified in SMPTE Standards 372M and 424M.

3.4 Stop (photographic)

A unit used to quantify ratios of light or exposure, with each added stop meaning a factor of two, and each subtracted stop meaning a factor of one-half.

3.5 Code value

In any integer encoding, any single value may be referred to as a code value, abbreviated as CV.

4 Specification

4.1 Naming conventions

Both the 10-bit and 12-bit logarithmic integer encoding of ACES specified in Section 4.3 and 4.4 of this document shall be known as ACESproxy.

Systems that are limited to the display of 8 characters for control labels shall use the abbreviation ACESPRXY.

Systems that are limited to the display of 5 to 7 characters for control labels shall use the abbreviation ACSPX.

4.2 Color space chromaticities

ACESproxy uses a different set of primaries than ACES RGB primaries defined in SMPTE ST 2065-1. The CIE 1931 colorimetry of the ACESproxy RGB primaries and white are specified below.

4.2.1 Color primaries

The RGB primaries chromaticity values, known as AP1, shall be those found in Table 1.

	R	G	B	CIE x	CIE y
Red	1.00000	0.00000	0.00000	0.713	0.293
Green	0.00000	1.00000	0.00000	0.165	0.830
Blue	0.00000	0.00000	1.00000	0.128	0.044

Table 1 – ACESproxy RGB primaries chromaticity values

4.2.2 White Point

The white point shall be that found in Table 2.

	R	G	B	CIE x	CIE y
White	1.00000	1.00000	1.00000	0.32168	0.33767

Table 2 – ACES RGB white point chromaticity values

4.3 ACESproxy 10-bit definition

The following functions shall be used to convert between ACES values, encoded according to SMPTE ST 2065-1, and 10-bit integer ACESproxy values.

4.3.1 10-bit ACESproxy encoding

ACES R, G, and B values shall be converted to lin_{AP1} R, G, and B values using the transformation matrix (TRA_1) calculated and applied using the methods provided in Section 4 of SMPTE RP 177:1993.

lin_{AP1} R, G, and B values shall be converted to ACESproxy10 values using Equation 1.

$$ACESproxy10 = \begin{cases} 64; & lin_{AP1} \leq 2^{-9.72} \\ \text{FLOAT2CV10}\left[(\log_2(lin_{AP1}) + 2.5) \times 50 + 425\right]; & lin_{AP1} > 2^{-9.72} \end{cases}$$

Where:

FLOAT2CV10($\mathbf{a}$) returns MAX(64, MIN(940, ROUND($\mathbf{a}$)))

ROUND($\mathbf{a}$) returns the integer value closest to the floating point value $\mathbf{a}$

MAX($\mathbf{a}, \mathbf{b}$) returns the greater of $\mathbf{a}$ or $\mathbf{b}$

MIN($\mathbf{a}, \mathbf{b}$) returns the lesser of $\mathbf{a}$ or $\mathbf{b}$

Equation 1 – $\mathbf{lin_{AP1}}$ to ACESproxy10

NOTE: Equation 2 shows the relationship between ACES R, G, and B values and lin_{AP1} R, G, and B values. TRA_1, rounded to 10 significant digits, is derived from the product of NPM_{AP1} inverse and NPM_{AP0} calculated using methods provided in Section 3.3 of SMPTE RP 177:1993. AP0 are the primaries of ACES specified in SMPTE ST 2065-1:2012. AP1 are the primaries of ACESproxy specified in Section 4.2.

$$\begin{bmatrix} R_{lin_{AP1}} \\ G_{lin_{AP1}} \\ B_{lin_{AP1}} \end{bmatrix} = TRA_1 \cdot \begin{bmatrix} R_{ACES} \\ G_{ACES} \\ B_{ACES} \end{bmatrix}$$

$$TRA_1 = \begin{bmatrix} 1.4514393161 & -0.2365107469 & -0.2149285693 \\ -0.0765537734 & 1.1762296998 & -0.0996759264 \\ 0.0083161484 & -0.0060324498 & 0.9977163014 \end{bmatrix}$$

$$TRA_1 = NPM_{AP1}^{-1} \cdot NPM_{AP0}$$

Equation 2 – ACES to $\mathbf{lin_{AP1}}$

NOTE 2: ACESproxy values encoded using the equation above are not appropriate for storage or for archiving. They are intended for use only with digital transport interfaces unable to carry half-precision floating-point values, and with integer-based grading systems designed to generate look metadata that will guide the color grading applied to ACES image data later in the post-production process.

NOTE 3: ACESproxy encodes values into the SMPTE "legal-range" for video systems; grading systems should use this as their nominal range.

4.3.2 10-bit ACESproxy decoding

ACESproxy R, G, and B values shall be converted to lin_{AP1} values using Equation 3.

$$lin_{AP1} = 2^{\left(\frac{ACESproxy10 - 425}{50} - 2.5\right)}$$

Equation 3 – ACESproxy10 to $\mathbf{lin_{AP1}}$

lin_{AP1} R, G, and B values shall be converted to ACES R, G, and B values using the transformation matrix (TRA_2) calculated and applied using the methods provided in Section 4 of SMPTE RP 177:1993.

NOTE: Equation 4 shows the relationship between ACES R, G, and B values and lin_{AP1} R, G, and B values. TRA_2, rounded to 10 significant digits, is derived from the product of NPM_{AP0} inverse and NPM_{AP1} calculated using methods provided in Section 3.3 of SMPTE RP 177:1993. AP0 are the primaries of ACES specified in SMPTE ST 2065-1:2012. AP1 are the primaries of ACESproxy specified in Section 4.2.

$$\begin{bmatrix} R_{ACES} \\ G_{ACES} \\ B_{ACES} \end{bmatrix} = TRA_2 \cdot \begin{bmatrix} R_{lin_{AP1}} \\ G_{lin_{AP1}} \\ B_{lin_{AP1}} \end{bmatrix}$$

$$TRA_2 = \begin{bmatrix} 0.6954522414 & 0.1406786965 & 0.1638690622 \\ 0.0447945634 & 0.8596711185 & 0.0955343182 \\ -0.0055258826 & 0.0040252103 & 1.0015006723 \end{bmatrix}$$

$$TRA_2 = NPM_{AP0}^{-1} \cdot NPM_{AP1}$$

Equation 4 – $\mathbf{lin_{AP1}}$ to ACES

4.4 ACESproxy 12-bit definition

The following functions shall be used to convert between ACES values, encoded according to SMPTE ST 2065-1, and 12-bit integer ACESproxy values.

4.4.1 12-bit ACESproxy encoding

ACES R, G, and B values shall be converted to lin_{AP1} R, G, and B values using the transformation matrix (TRA_1) calculated and applied using the methods provided in Section 4 of SMPTE RP 177:1993.

lin_{AP1} R, G, and B values shall be converted to ACESproxy12 values using Equation 5.

$$ACESproxy12 = \begin{cases} 256; & lin_{AP1} \leq 2^{-9.72} \\ \text{FLOAT2CV12}\left[(\log_2(lin_{AP1}) + 2.5) \times 200 + 1700\right]; & lin_{AP1} > 2^{-9.72} \end{cases}$$

Where:

FLOAT2CV12($\mathbf{a}$) returns MAX(256, MIN(3760, ROUND($\mathbf{a}$)))

ROUND($\mathbf{a}$) returns the integer value closest to the floating point value $\mathbf{a}$

MAX($\mathbf{a}, \mathbf{b}$) returns the greater of $\mathbf{a}$ or $\mathbf{b}$

MIN($\mathbf{a}, \mathbf{b}$) returns the lesser of $\mathbf{a}$ or $\mathbf{b}$

Equation 5 – $\mathbf{lin_{AP1}}$ to ACESproxy12

NOTE: Equation 6 shows the relationship between ACES R, G, and B values and lin_{AP1} R, G, and B values. TRA_1, rounded to 10 significant digits, is derived from the product of NPM_{AP1} inverse and NPM_{AP0} calculated using methods provided in Section 3.3 of SMPTE RP 177:1993. AP0 are the primaries of ACES specified in SMPTE ST 2065-1:2012. AP1 are the primaries of ACESproxy specified in Section 4.2.

$$\begin{bmatrix} R_{lin_{AP1}} \\ G_{lin_{AP1}} \\ B_{lin_{AP1}} \end{bmatrix} = TRA_1 \cdot \begin{bmatrix} R_{ACES} \\ G_{ACES} \\ B_{ACES} \end{bmatrix}$$

$$TRA_1 = \begin{bmatrix} 1.4514393161 & -0.2365107469 & -0.2149285693 \\ -0.0765537734 & 1.1762296998 & -0.0996759264 \\ 0.0083161484 & -0.0060324498 & 0.9977163014 \end{bmatrix}$$

$$TRA_1 = NPM_{AP1}^{-1} \cdot NPM_{AP0}$$

Equation 6 – ACES to $\mathbf{lin_{AP1}}$

NOTE 2: ACESproxy values encoded using the equation above are not appropriate for storage or for archiving. They are intended for use only with digital transport interfaces unable to carry half-precision floating-point values, and with integer-based grading systems designed to generate look metadata that will guide the color grading applied to ACES image data later in the post-production process.

NOTE 3: ACESproxy encodes values into the SMPTE "legal-range" for video systems; grading systems should use this as their nominal range.

4.4.2 12-bit ACESproxy decoding

ACESproxy R, G, and B values shall be converted to lin_{AP1} values using Equation 7.

$$lin_{AP1} = 2^{\left(\frac{ACESproxy12 - 1700}{200} - 2.5\right)}$$

Equation 7 – ACESproxy12 to $\mathbf{lin_{AP1}}$

lin_{AP1} R, G, and B values shall be converted to ACES R, G, and B values using the transformation matrix (TRA_2) calculated and applied using the methods provided in Section 4 of SMPTE RP 177:1993.

NOTE: Equation 8 shows the relationship between ACES R, G, and B values and lin_{AP1} R, G, and B values. TRA_2, rounded to 10 significant digits, is derived from the product of NPM_{AP0} inverse and NPM_{AP1} calculated using methods provided in Section 3.3 of SMPTE RP 177:1993. AP0 are the primaries of ACES specified in SMPTE ST 2065-1:2012. AP1 are the primaries of ACESproxy specified in Section 4.2.

$$\begin{bmatrix} R_{ACES} \\ G_{ACES} \\ B_{ACES} \end{bmatrix} = TRA_2 \cdot \begin{bmatrix} R_{lin_{AP1}} \\ G_{lin_{AP1}} \\ B_{lin_{AP1}} \end{bmatrix}$$

$$TRA_2 = \begin{bmatrix} 0.6954522414 & 0.1406786965 & 0.1638690622 \\ 0.0447945634 & 0.8596711185 & 0.0955343182 \\ -0.0055258826 & 0.0040252103 & 1.0015006723 \end{bmatrix}$$

$$TRA_2 = NPM_{AP0}^{-1} \cdot NPM_{AP1}$$

Equation 8 – $\mathbf{lin_{AP1}}$ to ACESproxy

Appendix A

(informative)

Viewing of ACESproxy images

As a part of the ACES system, images encoded in ACESproxy form are intended to be decoded into ACES values and viewed using the Reference Rendering Transform (RRT) and an Output Device Transform (ODT) appropriate for an intended viewing device.

Without such a transform in place, viewed ACESproxy images will appear dim, severely low in contrast and desaturated. However, directly viewing the unrendered log-encoded images is sometimes useful, for example while looking at the wide range of captured image data in the highlights and shadows that are preserved in the ACES system.

ACESproxy has been designed to place scene details into the SMPTE "legal-range" of video systems. Scene detail from about 7 stops under mid-gray to 10 stops over mid-gray should be visible within normal legal-range monitor setups. No rescaling of the device output signal should be needed for direct viewing, but is required before applying color grading transforms as described in Appendix C.

The ACESproxy encoding allows an amount of 'headroom' beyond the current dynamic range capabilities of digital motion picture cameras, and it is expected that the range of exposed highlight values seen on a waveform monitor will be lower on a monitors scale than the corresponding range that would be shown if other forms of log encoding were used.

Specific knowledge of the dynamic range of a camera system and its output encoding can be used to determine the maximum value that will appear on a waveform monitor indicating an exposure has reached full saturation of the sensor.

On a waveform monitor displaying ACESproxy values in IRE units, a gray card representing an 18% reflectance would appear at a level of 41% IRE under a normal exposure assumption. A perfect white reflector under the same conditions would appear at 55% IRE. A camera which reaches sensor saturation at 7 stops above 18% reflectance would not show any values above 81% IRE.

Appendix B

(informative)

Range of ACES values

This appendix is intended for developers who wish to validate the accuracy of their implementation.

The table below contains the results of conversions using exact 16-bit ACES codes. 16-bit ACES has higher precision than either form of ACESproxy so rounding of ACES values will occur. These numbers are accurate for neutral values where R=G=B.

ACES 16-bit half-float Hex Code	ACES	ACESproxy 10-bit CV	ACESproxy 12-bit CV	ACES decoded from 10-bit ACESproxy	ACES decoded from 12-bit ACESproxy
14DA	0.001184464	64	256	0.001185417	0.001185417
31C3	0.180053711	426	1705	0.179199219	0.179809570
5AF7	222.875	940	3760	222.875	222.875

Appendix C

(informative)

Convention for use of ACESproxy SMPTE range

This appendix defines the equivalent of a [0.0 ... 1.0] range in both 10-bit and 12-bit ACESproxy for use in applications.

For consistency in using ACESproxy values in video systems designed around the [0.0 ... 1.0] video range (encoded in 10-bit video as the range 64-940), the following ACES values are considered the equivalents to 0.0 and 1.0 for both 10-bit and 12-bit video.

Video normalized scale	IRE	10-bit CV	12-bit CV	ACES
0.0	0%	64	256	0.001185417
1.0	100%	940	3760	222.875

ASC CDL values are applied to values in the range [0.0 ... 1.0] which by this convention are set to ACES values 0.001185417 to 222.875. In integer based color grading systems this is typically accomplished by scaling from "legal" to full range before applying ASC CDL transforms. The ASC CDL application is further defined in a separate document.

Appendix D

(informative)

ACESproxy function derivation

The ACESproxy 10-bit and 12-bit logarithmic encoding and decoding functions have been derived from the single mathematical function described below. A series of parameters are defined and the values for the parameters specified based on the bit depth of the encoding.

Math functions

The following general-use math functions are defined for use within the equations.

ROUND($\mathbf{a}$) Math function taking a floating-point value $\mathbf{a}$, and returning the integer value closest to $\mathbf{a}$.

MAX($\mathbf{a}, \mathbf{b}$) Math function returning the greater of $\mathbf{a}$ or $\mathbf{b}$

MIN($\mathbf{a}, \mathbf{b}$) Math function returning the lesser of $\mathbf{a}$ or $\mathbf{b}$

FLOAT2CV($\mathbf{a}$) Math function returning MAX($CVmin$, MIN($CVmax$, ROUND($\mathbf{a}$)))

Parameters

The following parameters are defined for each bit-depth.

- $CVmin$ is the minimum code value available for representation of ACES image data.
- $CVmax$ is the maximum code value available for representation of ACES image data.
- $StepsPerStop$ is the number of code values representing a change of 1 stop in exposure.
- $MidCVoffset$ is the integer code value representing the assigned midpoint of the exposure scale for a particular bit-depth encoding. (e.g. the point to which a mid-grey exposure value would be mapped)
- $MidLogOffset$ is the base 2 logarithmic value representing the assigned midpoint of the exposure scale in log space, [e.g. $MidLogOffset = \log_2(2^{-2.5}) = -2.5$]

	ACESproxy 10-bit CV	**ACESproxy 12-bit CV**
$CVmin$	64	256
$CVmax$	940	3760
$StepsPerStop$	50	200
$MidCVoffset$	425	1700
$MidLogOffset$	-2.5	-2.5

NOTE: $MidCVoffset$ is not equal to the ACESproxy value that most closely represents an ACES mid-gray value of 0.18. ACES 0.18 is most closely represented by ACESproxy 426 10-bit CV and 1705 12-bit CV.

Encoding Function

The following floating-point equation is used to convert linear values to integer code values.

$$ACESproxy = \begin{cases} CVmin; & lin \leq 2^{\left(\frac{(CVmin - MidCVoffset)}{StepsPerStop} - MidLogOffset\right)} \\ \text{FLOAT2CV}\left[(\log_2(lin) - MidLogOffset) \times StepsPerStop + MidCVOffset\right]; & lin > 2^{\left(\frac{(CVmin - MidCVoffset)}{StepsPerStop} - MidLogOffset\right)} \end{cases}$$

where $ACESproxy$ is the resulting integer code value in the range of code values from $CVmin$ to $CVmax$.

An implementation may use mathematically equivalent forms of this encoding equation.

Decoding Function

The following floating-point equation is used to convert ACESproxy integer code values to linear values.

$$lin = 2^{\left(\frac{(ACESproxy - MidCVoffset)}{StepsPerStop} + MidLogOffset\right)}$$

The conversion to linear creates the closest value in 16-bit half precision floating-point to the floating-point result of the equation. Linear values resulting from this equation are limited to the range of values that can be encoded in ACESproxy as illustrated in Appendix B. This decoding function does not produce negative values.

An implementation may use mathematically equivalent forms of this decoding equation.