

SMPTE STANDARD

ACES Image Container File Layout



Page 1 of 36 pages

Table of Contents	Page
Foreword	5
Intellectual Property	5
Introduction.....	5
1 Scope	6
2 Conformance Notation	6
3 Normative References	6
4 Terms and Definitions	7
5 General.....	9
5.1 Image	9
5.2 Image Channels	9
5.3 Alpha Channel.....	9
5.4 Display Window, Data Window	10
5.5 Scan Lines	11
5.6 Screen Window	11
6 Basic Data Types	12
6.1 Grouping	12
6.2 Packing.....	12
6.3 Integer Numbers	13
6.4 Floating-Point Numbers	13
6.5 Character Encoding, Sequences, and Strings.....	13
7 File Layout.....	14
7.1 File Extension.....	14
7.2 High-Level File Layout	14
7.3 Magic Number	14
7.4 Version Field	14
7.5 Header.....	15
7.6 Line Offset Table.....	17
7.7 Scan Line Storage.....	17
7.8 End-of-File Filler.....	18
8 Predefined Attribute Types	18
8.1 General.....	18
8.2 box2i.....	18
8.3 chlist	18
8.4 chromaticities	19
8.5 compression.....	19
8.6 double.....	19

8.7	float	19
8.8	half	19
8.9	int	19
8.10	lineOrder	20
8.11	keycode	20
8.12	rational	20
8.13	short	21
8.14	string	21
8.15	stringVector	21
8.16	timecode	21
8.17	unsignedChar	23
8.18	unsignedInt	23
8.19	unsignedLong	23
8.20	unsignedShort	23
8.21	v2f	23
8.22	v3f	23
9	Predefined Attributes	24
9.1	General	24
9.2	acesImageContainerFlag	24
9.3	altitude	24
9.4	aperture	24
9.5	cameraFirmwareVersion	24
9.6	cameraIdentifier	24
9.7	cameraLabel	24
9.8	cameraMake	24
9.9	cameraModel	25
9.10	cameraPosition	25
9.11	cameraSerialNumber	25
9.12	cameraUpDirection	25
9.13	cameraViewingDirection	25
9.14	capDate	26
9.15	captureRate	26
9.16	channels	26
9.17	chromaticities	27
9.18	comments	27
9.19	compression	27
9.20	convergenceDistance	27
9.21	creator	27
9.22	dataWindow	28
9.23	displayWindow	28
9.24	expTime	28
9.25	focalLength	28
9.26	focus	28
9.27	framesPerSecond	28
9.28	free	28
9.29	imageDigestMD5	28
9.30	imageCounter	28
9.31	imageRotation	29
9.32	interocularDistance	29
9.33	isoSpeed	29
9.34	keyCode	29
9.35	latitude	29
9.36	lensAttributes	29
9.37	lensMake	29
9.38	lensModel	29

9.39	lensSerialNumber.....	29
9.40	lineOrder	30
9.41	longitude.....	30
9.42	metadataDigestMD5	30
9.43	multiView.....	30
9.44	originalImageFlag.....	30
9.45	owner.....	30
9.46	pixelAspectRatio	30
9.47	recorderFirmwareVersion.....	30
9.48	recorderMake	31
9.49	recorderModel	31
9.50	recorderSerialNumber.....	31
9.51	reelName.....	31
9.52	screenWindowCenter.....	31
9.53	screenWindowWidth	31
9.54	storageMediaSerialNumber	31
9.55	timeCode.....	31
9.56	timecodeRate	31
9.57	utcOffset.....	31
9.58	uuid.....	31
10	Attributes for Stereoscopic Images	32
11	Reader and Writer Requirements	32
Annex A	Bibliography (Informative)	33
Annex B	Sample File (Informative)	34

List of Tables

Table 1 – Integer Data Types	13
Table 2 – Floating Point Data Types	13
Table 3 – File Structure	14
Table 4 – Attribute Structure	15
Table 5 – Required Attributes	16
Table 6 – Scan Line Object	17
Table 7 – box2i Structure	18
Table 8 – chlist Structure	19
Table 9 – chromaticities Structure	19
Table 10 – keyCode Structure and Values	20
Table 11 – rational Structure	20
Table 12 – timecode Structure	21
Table 13 – timeAndFlags Structure	22
Table 14 – userData Structure	22
Table 15 – v2f Structure	23
Table 16 – v3f Structure	23
Table 17 – chromaticities Values	27
Table 18 – Names Of Stereoscopic Attributes	32

Foreword

SMPTE (the Society of Motion Picture and Television Engineers) is an internationally-recognized standards developing organization. Headquartered and incorporated in the United States of America, SMPTE has members in over 80 countries on six continents. SMPTE's Engineering Documents, including Standards, Recommended Practices, and Engineering Guidelines, are prepared by SMPTE's Technology Committees. Participation in these Committees is open to all with a bona fide interest in their work. SMPTE cooperates closely with other standards-developing organizations, including ISO, IEC and ITU.

SMPTE Engineering Documents are drafted in accordance with the rules given in Part XIII of its Operations Manual.

SMPTE ST 2065-4 was prepared by Technology Committee 31FS.

Intellectual Property

At the time of publication, no notice had been received by SMPTE claiming patent rights essential to the implementation of this standard. However, attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. SMPTE shall not be held responsible for identifying any or all such patent rights.

Introduction

The ACES image container is a digital data file format that is designed for storing the individual frames of a motion picture during production and for archival. During production, it is important to maintain the highest possible image quality, and to minimize generational loss as images are repeatedly extracted from files, processed, and stored in new files. Fast random access to individual frames is highly desirable. Storage space and data transmission bandwidth limitations tend to be of secondary concern.

The ACES image container is not meant for distribution of motion pictures to theaters and home viewers, where representation of an entire movie as a single file, compact storage and data security take precedence.

Each ACES image container file contains a single monoscopic or stereoscopic image. Moving images are represented as sequences of image container files, with a separate image container file for each frame. This allows each frame to be read or written individually, without accessing other frames or any kind of enclosing image sequence container.

The ACES image container file format is derived from the popular OpenEXR file format, and is readable by many OpenEXR readers.

1 Scope

This Standard specifies the layout of, and metadata in, files containing images that conform to SMPTE ST 2065-1, Academy Color Encoding Specification (ACES).

2 Conformance Notation

Normative text is text that describes elements of the design that are indispensable or contains the conformance language keywords: "shall", "should", or "may". Informative text is text that is potentially helpful to the user, but not indispensable, and can be removed, changed, or added editorially without affecting interoperability. Informative text does not contain any conformance keywords.

All text in this document is, by default, normative, except: the Introduction, any section explicitly labeled as "Informative" or individual paragraphs that start with "NOTE" or "EXAMPLE"

The keywords "shall" and "shall not" indicate requirements strictly to be followed in order to conform to the document and from which no deviation is permitted.

The keywords, "should" and "should not" indicate that, among several possibilities, one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required; or that (in the negative form) a certain possibility or course of action is deprecated but not prohibited.

The keywords "may" and "need not" indicate courses of action permissible within the limits of the document.

The keyword "reserved" indicates a provision that is not defined at this time, shall not be used, and may be defined in the future. The keyword "forbidden" indicates "reserved" and in addition indicates that the provision will never be defined in the future.

A conformant implementation according to this document is one that includes all mandatory provisions ("shall") and, if implemented, all recommended provisions ("should") as described. A conformant implementation need not implement optional provisions ("may") and need not implement them as described.

Unless otherwise specified, the order of precedence of the types of normative information in this document shall be as follows: Normative prose shall be the authoritative definition; Tables shall be next; followed by formal languages; then figures; and then any other language forms.

3 Normative References

Note: All references in this document to other SMPTE documents use the current numbering style (e.g. SMPTE ST 12-1:2008) although, during a transitional phase, the document as published (printed or PDF) may bear an older designation (such as SMPTE 12M-1-2008). Documents with the same root number (e.g. 12-1) and publication year (e.g. 2008) are functionally identical.

The following standards contain provisions that, through reference in this text, constitute provisions of this standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent edition of the standards indicated below.

IEEE Std 754-2008, Standard for Floating-Point Arithmetic

IETF RFC 1321:1992, The MD5 Message-Digest Algorithm (Proposed Standard)

IETF RFC 4122:2005, A Universally Unique Identifier (UUID) URN Namespace (Proposed Standard)

ISO 11664-1:2007 (CIE S 014-1/E:2006), Colorimetry — Part 1: CIE Standard Colorimetric Observers

SMPTE ST 12-1:2008, Television — Time and Control Code

SMPTE ST 254:2008, Motion-Picture Film (35-mm) — Manufacturer-Printed Latent Image Identification Information

SMPTE ST 268:2003, File Format for Digital Moving-Picture Exchange (DPX), Version 2.0

SMPTE ST 331:2011, Element and Metadata Definitions for the SDTI-CP

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

The Unicode Consortium. The Unicode Standard, Version 6.0.0, (Mountain View, CA: The Unicode Consortium, 2011. ISBN 978-1-936213-01-6)

4 Terms and Definitions

For the purposes of this document, the following terms and definitions apply.

4.1

ACES image container file

file compliant with this standard

4.2

alpha opacity

proportion by which a pixel obscures the pixel behind it (from ISO/IEC 9592-1:1997)

4.3

byte order

ordering of bytes for multi-byte data primitives within a file, such as 4-byte integers, or 8-byte floats

4.4

channel

collection of all values of the same name in an image or in a part of an image

4.5

compositing

combining two images into a single image (from ISO/IEC 15444-2:2004)

4.6

editor

device that is both a compliant reader and a compliant writer and that updates files, or reads files and writes new files with derived contents

4.7

exposure time

time from the start of light exposure integration at a point on the sensor to the end of light exposure integration at that same point on the sensor

4.8

little-endian byte order

byte order having increasing numerical significance with increasing byte address (from the least significant byte to the most significant byte)

4.9

monoscopic image

image containing a single view of the depicted scene, intended to be presented to the viewer on a 2D display device

4.10

"over" compositing operation

compositing, where a foreground RGB color F with an associated alpha value A over a background RGB color B , will produce a combined color C , where $C = F + (1-A) * B$ (from Porter-Duff, 1984)

4.11

pixel aspect ratio

pixel width divided by pixel height when the image is displayed with the intended aspect ratio, where the pixel width is the distance between the centers of two horizontally adjacent pixels on the display, and the pixel height is the distance between the centers of two vertically adjacent pixels on the display

4.12

pixel space

two-dimensional Cartesian coordinate system having signed integer coordinates with x coordinates increasing from left to right and y coordinates increasing from top to bottom

4.13

reader

device consuming files compliant with this standard

4.14

recorder

device that stores image data and image metadata received from a digital capture system

4.15

RGB

color model with red, green, blue components

4.16

scan line

row of pixels, with all pixels having the same y coordinate

4.17

stereoscopic image

image containing two separate views of the depicted scene, intended to be presented to the viewer on a 3D display device such that the left-eye view and right-eye view are visible only to the viewer's left and right eye respectively

4.18

window

axis-parallel rectangular region in pixel space, specified by the coordinates of two opposing corners, $(x_{\min}, y_{\min})$ and $(x_{\max}, y_{\max})$, where $x_{\min} < x_{\max}$ and $y_{\min} < y_{\max}$, and including all pixels with coordinates (x, y) where $x_{\min} \leq x \leq x_{\max}$, and $y_{\min} \leq y \leq y_{\max}$

4.19

writer

device producing files compliant with this standard

5 General

5.1 Image

The file shall contain exactly one image.

The image shall be an RGB image, monoscopic or stereoscopic, with or without alpha channels. Section 5.2 specifies, for each type of image, the channels that are present in the file.

The red, green, and blue components of pixels shall be as defined in SMPTE ST-2065-1 (Academy Color Encoding Specification).

The alpha value of a pixel shall be as defined in Section 5.3.

The image shall contain one value for each channel of each pixel in the data window.

The red, green, blue, and alpha values are of type `half` (16-bit floating-point).

Section 7.7 describes how pixel data and channels are organized into scan lines.

5.2 Image Channels

The type of image determines the channels present in the file:

- A monoscopic image without alpha shall contain channels “B”, “G”, and “R”.
- A monoscopic image with alpha shall contain channels “A”, “B”, “G”, and “R”.
- A stereoscopic image without alpha shall contain channels “B”, “G”, “R”, “left.B”, “left.G”, and “left.R”.
- A stereoscopic image with alpha shall contain channels “A”, “B”, “G”, “R”, “left.A”, “left.B”, “left.G” and “left.R”.

Note: OpenEXR allows an arbitrary number of image channels. The ACES image container restricts the set of image channels that can appear in a file to the four possibilities listed above.

Channels “B”, “G”, and “R” shall represent the blue, green, and red components of the pixel of the monoscopic image or in the right-eye view.

EXAMPLE 1: The collection of all R values of all pixels in a scan line is the “R channel” of that scan line.

Channels “left.B”, “left.G” and “left.R” shall represent the blue, green, and red components of the pixel in the left-eye view.

Channel “A” shall represent alpha of the pixel of the monoscopic image or in the right-eye view.

Channel “left.A” shall represent alpha of the pixel in the left-eye view.

The channel order in scan lines and in the channels attribute shall be “A”, “B”, “G”, “R”, “left.A”, “left.B”, “left.G”, “left.R”.

5.3 Alpha Channel

When an Alpha channel is not present, the pixels shall be considered fully opaque.

When an Alpha channel is present, the alpha value in the pixel's alpha channel shall specify the proportion by which the pixel shall obscure the background image's pixel if used in an "over" compositing operation. This is known as the opacity of the pixel.

The alpha values shall be in the range [0.0, 1.0]. A value of 0.0 shall indicate that the pixel is fully transparent, and 1.0 shall indicate that the pixel is fully opaque.

The pixel's RGB values shall specify the amount of color that the pixel shall contribute if used in an "over" compositing operation.

The RGB values may exceed the alpha value, so the pixel can contribute color to the composition also when the alpha value is zero.

Note 1: While the above definition looks like "premultiplied alpha", this term is not used here, as some definitions of "premultiplied alpha" or "associated alpha" require that all RGB values are premultiplied with alpha or disallow $RGB > alpha$. For compatibility with OpenEXR, this file format does not impose such requirements on RGB. Other documents can contain workflow-specific requirements on preparing RGB values prior to storing them in the file. The above definition is not related to "straight alpha" or "unassociated alpha".

Note 2: $RGB > alpha$ is useful for semi-transparent sources of light, such as fire with semi-transparent smoke, head-up displays, glow, lens flare, or window reflections.

Note 3: For color corrections and other operations that work on "straight alpha" instead of "premultiplied alpha", any premultiplied alpha baked into the file's RGB values can be removed by dividing the RGB values with alpha. As the RGB values are in floating point, this can be done without loss of precision over a wide range of values, except for $alpha = 0$. This removal is not appropriate when alpha is not baked into the file's RGB values, such as for fire, in which case other adjustments (if any) can be more appropriate.

5.4 Display Window, Data Window

Two windows specify the boundaries of the image:

The *display window* shall define the window that is to be displayed.

The *data window* shall define the window for which pixel data are available in the file.

No specific spatial relationship between the data window and the display window is required. The two windows may be coincident, they may overlap, or they may be completely disjoint.

EXAMPLES: Assume that a motion picture is produced with a resolution of 2048 by 1080 pixels. The upper left and lower right corners of the display window for all frames of the movie are at (0, 0) and (2047, 1079). For most images, in particular finished frames that will be part of the final product, the data window is the same as the display window, but for some images that are used in producing the finished frames, the data window differs from the display window.

For a background plate that will be heavily post-processed, extra pixels beyond the edge of the display window are recorded. The upper left and lower right corners of the data window are at (-100, -100) and (2147, 1179). The extra pixels are not normally displayed. Their existence allows operations such as large-kernel blurs or simulated camera shake to avoid edge artifacts.

Alternatively, while working on a computer-generated element, an artist can repeatedly render the same frame. To save time, the artist can render only a small region of interest, and the resulting image files contain no pixel data outside this region of interest. When the image is displayed, the area that is inside the display window but outside of the data window can be filled with an arbitrary color.

5.5 Scan Lines

The pixels in the data window shall be stored as *scan lines*. A data window with opposing corners $(x_{\min}, y_{\min})$ and $(x_{\max}, y_{\max})$ shall contain

$$y_{\max} - y_{\min} + 1$$

scan lines. Each scan line shall contain

$$x_{\max} - x_{\min} + 1$$

pixels.

5.6 Screen Window

The *screen window* is the boundary (left-right and top-bottom) of the camera's view of the scene, and it establishes a relationship between the camera's view of the scene and the file's display window. The screen window is a rectangle in a screen window coordinate system.

The screen window coordinate system shall be a left-handed, Cartesian coordinate system.

Its x-axis shall point in the same direction (to the right) as the x-axis in the display window (which is in pixel space).

Its y-axis shall point in the opposite direction (up) of the y-axis in the display window.

Its z-axis shall point in the viewing direction, into the scene.

The camera shall be at location $\{0, 0, 0\}$ and shall look along the positive z axis.

The screen window shall be an axis-aligned rectangle on the $z = +1$ plane.

The screen window's rectangle shall map to the display window's rectangle (the latter in pixel space).

The camera's view of the scene as shown in the display window is then the scene cropped by the screen window boundary at the $z = +1$ plane.

The screen window's rectangle is specified by its width and center position.

The screen window's width W shall be the width of the rectangle in the screen window coordinate system.

The screen window's center C shall be the x and y coordinates of the rectangle's center in the screen window coordinate system.

Note 1: The screen window height H can be derived from the screen window width W , display window width dW and height dH and the pixel aspect ratio pR as follows:

$$H = W \, dH / dW / pR$$

The coordinates of the corners of the screen window / display window can be computed from C , W and H :

$$\{x, y\} = \{C.x \pm W / 2, C.y \pm H / 2\}$$

Note 2: The screen window coordinate system is different from the coordinate systems used in Section 9.13 (cameraViewingDirection) and Section 9.10 (cameraPosition). The screen window coordinate system is locked with the pixel space and is dimensionless, so the $z = +1$ plane can be thought of as being placed at any distance from the camera including behind the scene.

EXAMPLE: If $C = \{0, 0\}$ and $W = 1$, then the horizontal viewing angle is 53 degrees, and the camera is aligned with the center of the image.