OpenColorIO

and its usage with The Academy Color Encoding Specification (ACES) in a modern visual effects pipeline

Bachelorarbeit im Studiengang Audiovisuelle Medien

Hochschule der Medien Stuttgart

vorgelegt von

Falk Lude

Matrikelnummer: 26855

Erstprüfer: **Prof. Katja Schmid**

Zweitprüfer: **Falk Hofmann**

Stuttgart, 05. September 2016

OpenColorIO

and its usage with The Academy Color Encoding Specification (ACES) in a modern visual effects pipeline

Bachelor thesis

submitted to the department of Electronic Media

Media University Stuttgart

in partial fulfillment of the requirements for the degree of Bachelor of Engineering

by

Falk Lude

Student ID: 26855

First examiner: **Prof. Katja Schmid**

Second examiner: **Falk Hofmann**

Stuttgart, September 2016

Ehrenwörtliche Versicherung

„Hiermit versichere ich, Falk Lude, ehrenwörtlich, dass ich die vorliegende Bachelorarbeit mit dem Titel: ***„OpenColorIO and its usage with The Academy Color Encoding Specification (ACES) in a modern visual effects pipeline“*** selbstständig und ohne fremde Hilfe verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe. Die Stellen der Arbeit, die dem Wortlaut oder dem Sinn nach anderen Werken entnommen wurden, sind in jedem Fall unter Angabe der Quelle kenntlich gemacht. Die Arbeit ist noch nicht veröffentlicht oder in anderer Form als Prüfungsleistung vorgelegt worden.

Ich habe die Bedeutung der ehrenwörtlichen Versicherung und die prüfungsrechtlichen Folgen (§26 Abs. 2 Bachelor-SPO (6 Semester), § 24 Abs. 2 Bachelor-SPO (7 Semester), § 23 Abs. 2 Master-SPO (3 Semester) bzw. § 19 Abs. 2 Master-SPO (4 Semester und berufsbegleitend) der HdM) einer unrichtigen oder unvollständigen ehrenwörtlichen Versicherung zur Kenntnis genommen.“

Ort, Datum Unterschrift

Abstract

With the establishment of digital cameras used in motion pictures, the variety of different models grows continuously. Nowadays, the usage of more than one camera in a motion picture is not unusual. Because of different sensor types and color spaces and the result of color differences, it is very important to use a strong color management solution in the post production.

With the open source software ***OpenColorIO (OCIO),*** it is possible to create a clear and powerful color workflow which can be used in most of the common software packages.

This thesis gives a detailed overview in OpenColorIO and focusses on the scope in a modern visual effects pipeline. Furthermore, it explores the ***Academy Color Encoding Specification (ACES)*** which is integrated in OpenColorIO.

Kurzfassung

Mit der Etablierung der Digitalen Filmkamera im Spielfilmbereich nimmt deren Vielfalt an unterschiedlichen Modellen kontinuierlich zu. In einem modernen Spielfilm ist es längst keine Ausnahme mehr, mit mehreren Kameras von unterschiedlichen Herstellern zu filmen. Durch oft verschiedene Sensoren oder unterschiedlicher Farbräume und die daraus resultierenden Farbunterschiede ist es in der heutigen Postproduktion wichtiger denn je, ein robustes Color Management zu benutzen.

Mit der Open Source Software ***OpenColorIO (OCIO)*** ist es möglich, einen übersichtlichen Color Workflow aufzubauen, der mit einer Vielzahl der üblichen Softwarepakete genutzt werden kann.

Diese Thesis gibt einen detaillierten Einblick in OpenColorIO mit Fokus auf den Anwendungsbereich in einer modernen Visual Effects Pipeline. In diesem Zusammenhang wird näher auf den in OpenColorIO integrierten ACES-Workflow eingegangen.

Contents

1 Introduction

Before I knew what I was going to write for my bachelor thesis, I had the chance to work as a Digital Imaging Technician (DIT) and VFX editor for an internal studio production, a short documentary called ***“Painting Doors”***[1], at the “***Hochschule der Medien” (HdM)***[2]. Because of many different requirements such as interviews, visual effect shots at night and day, slow motion shots in front of a green screen or shooting with a drone, we had to use five different digital cameras. As the DIT and VFX editor on this project, it was my job to bring all those different footages together for creating proxies and preparing the plates for the visual effect shots. During this project, I learned a lot about color management and how important it is, especially in bigger movie productions with more than just one facility involved.

Thus, I came up with the idea to write my bachelor thesis about a topic with color management as a fundamental point. Because I already work as a digital compositor in the visual effects industry at the time this thesis is written, I am going to focus on the visual effects pipeline aspects in color management. OpenColorIO is a color management solution with an emphasis on visual effects and computer animation. Nowadays, it is used by most of the big visual effects companies around the world. It is open source and can easily be implemented and modified in most of the common software packages because it is based on C++ and Python[3].

1.1 Motivation

The motivation is to give a detailed insight in color management during post production and take a deeper look into a modern visual effects pipeline. Additionally, an usual OCIO color workflow will be compared with the Academy Color Encoding Specification (ACES) workflow within OCIO to show the benefits of a strong color management and the right workflow in a visual effects pipeline.

1 cf. <http://www.imdb.com/title/tt5486448/?ref_=nm_knf_t4> (2016-07-20)
2 cf. <http://www.hdm-stuttgart.de> (2016-07-20)
3 cf. <http://opensource.imageworks.com/?p=opencolorio> (2016-07-20)

1.2 Target Audience

The thesis is written for people working in visual effects or post production facilities. This includes first of all VFX editors, colorists and pipeline developer, but also compositors, matte painters, cg artists and supervisors or producers. The thesis could be also interesting for people on the film set, such as technical directors (TDs), Digital Imaging Technicians (DITs) or Directors of Photography (DOPs).

The thesis assumes a basic knowledge of filmmaking and their terminology from the reader. Because of length constraints, not every technical aspect is explained in detail, but I will do my best that, at the end, the whole topic is combined in an understandable way.

1.3 Content

The thesis consists of five main parts beginning with basic topics of color science. After a short introduction to the physical perception of colors, the paper explains color spaces and the basic concepts of color transformations. In the last part of chapter 2, the scene-referred imagery is compared with the display-referred imagery.

The second main part starts with chapter 3 and is about the color management in a visual effects pipeline. First, a short overview of the post production workflow is given. After that, VFX pre-grades are described and why it is so important to recreate them correctly. Chapter 3 completes with describing common workflow problems which are often a result of bad communication between the production companies.

The third main part is focused on OpenColorIO. After explaining the concept of OpenColorIO, an overview of potential applications in a visual effects process, in which OCIO can be a useful tool, is given. After that, a basic overview of the software and its configuration syntax is described.

Chapter 5 describes the fourth main part of the thesis, the ACES workflow. The chapter starts with explaining what ACES contents and gives a detailed look at its workflow. After that, the standardized OpenEXR format is described. Chapter 5 completes with comparing the advantages and disadvantages of the ACES workflow.

The last main part of this paper shows some practical examples. An OCIO setup within the ACES workflow is configured and explained in detail. Paying special attention to the compositing workflow in The Foundry's Nuke[4] and the matte painting process in Adobe's Photoshop[5] in which the ACES color spaces *"ACEScg"* and *"ACEScc"* are used as working spaces.

The thesis finishes with chapter 7 in which I will give a conclusion about the topic and an outlook on further opportunities with OpenColorIO.

4 cf. <http://www.thefoundry.co.uk> (2016-07-20)

5 cf. <http://www.adobe.com/de/products/photoshop.html> (2016-08-02)

2 Color Science

Bringing the right colors on the screen of theaters or television is a complex process which has its beginning with the physical laws of colors and how to bring them through the camera into the digital world. For the understanding of the below paper it is essential to know some of the basic concepts of color science. Thus, this chapter is giving an introduction about the physical perception of colors and clarifies questions about color spaces and color transformation as well as the differences between scene-referred imagery and display-referred imagery.

2.1 Physical Perception of Colors

Capturing light in cameras follows the same physical principles as the eye of a human being. Therefore it is worth to take a look at the physical perception of colors and how it influenced the technical development of cameras and sensors. The measurement and characterization of colors is described as colorimetry. This knowledge is an important component of color management[6]. Jeremy Selan, the developer of OpenColorIO, describes in his paper ***"Cinematic Color"*** the importance of color science for color management as followed: *"Without color science, it would not be possible to characterize displays, characterize cameras, or have an understanding of the imaging fundamentals that permeate the rest of computer graphics."*[7]

The understanding of colors in a scientific aspect has its origin in the year 1666. Isaac Newton established, as the first person in history, experimental facts about the perception of colors[8]. In his experiment, he has cut a hole in his window shape and submitted sunlight through it. First, it strikes a lens to sheave the light, then it hits a prism which spreads the sunlight and produces an array of different colors on a card. With this experiment, Newton figured out, that in the sunlight are some components which let colors appear when separated by the prism. This strip of light is colored red, orange, yellow, green, blue, indigo and violet, Newton described it as the color spectrum. With another experiment in which he tried to separate a single band of the spectrum with a second prism he proved, that the spectral colors were the basic components of white light because they cannot be separated twice[9].

---

6 cf. Selan, 2012, p 7
7 Selan, 2012, p 7
8 cf. Hunt et al., 2011, p 1
9 cf. Hunt et al., 2011, p 2

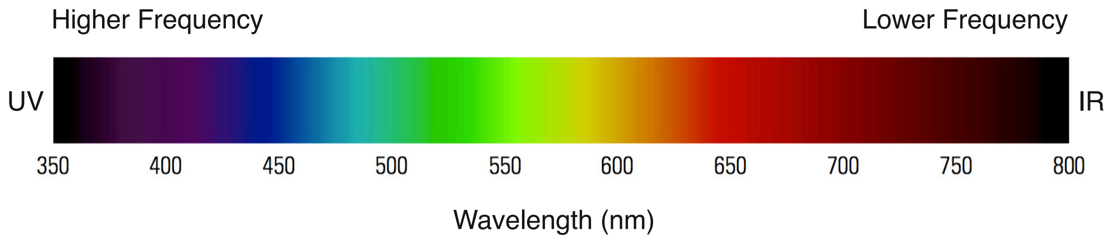


Figure 2.1: Colors of the electromagnetic spectrum, visible to human observers

The light energy consists of electro-magnetic radiation and can be divided in very short wavelengths[10] [11]. The most sensitive wavelengths that the human visual system can perceive is about 380-780 nm (yellow-green range). The more it comes to the extremes, the less colors are distinguishable[12].

In 1675, Newton wrote up the idea, that the rays of light are not colored but rather contain a certain power in them to save a specific sensation. What he wanted to say is, that color is in the eye and the brain and not in physics. This theory helped to define the fundamentals of vision which specifies that the human visual system is trichromatic (three types of cone cells in the retina of a human eye)[13]. Thus, the human vision can be divided in three parts of different wavelengths. Two commonly used designations were the L, M, S (Long, Medium and Short wavelength) and the R, G, B designation. In those models, L is used for luminance and represents red, M is responsible for the color fullness and represents green, and S is used for saturation and represents blue[14].

2.2 The CIE 1931 Color Matching Functions

With the knowledge, that the human vision is divided into three parts, the CIE[15] experimented in the 1920s to replicate the vision of the eye to a scanner. They tried to match the power amount of R, G, B to a constant level, without losing to much of the spectrum. In 1931, they standardized a set of functions and named these curves $\bar{x}$, $\bar{y}$, $\bar{z}$ functions (CMFs) for the CIE Standard Observer (illustrated in ***Figure 2.2***)[16].

10 Nanometre (abbreviation, nm)
11 cf. Hunt et al., 2011, p 2
12 cf. Selan, 2012, p 7
13 cf. Poynton, 2003, p 212
14 cf. Hunt et al., 2011, p 7
15 Commission Internationale de L'Eclaire
16 cf. Poynton, 2003, p 216

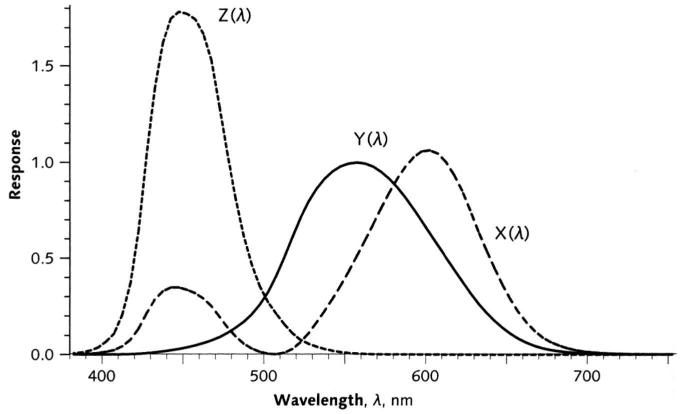


Figure 2.2: The CIE 1931 color-matching functions

2.3 Color Spaces

With the color-matching functions, it is possible to map a spectral power distribution (SPD)[17] into three numerical values (primaries), which define the mathematical coordinates of a color space, designated as ***CIE XYZ tristimulus values***[18]. To calculate these values, a spectral power distribution is integrated with the CIE 1931 curves. The resulting components were labeled *X*, *Y*, and *Z* (with capitalization)[19]. The additive mixture of the three primaries, also referred as "***color encoding***", reproduces a range of colors, which is defined as the gamut. It defines the displayable colors in a specific color space. One of the most important set of primaries in digital post production is the Rec.709, which is the international standard for HDTV displays[20]. This is a display-referred color space, which is described in more detail later in this chapter.

2.3.1 The CIE [x, y] Chromaticity Diagram

For a visual understanding and computation purposes it is convenient to separate the lightness from the colors. Therefore, the CIE developed a standardized procedure for normalizing the XYZ tristimulus values to get two chromaticity values, defined as *x* and *y*.

17 A spectral power distribution expresses the physical wavelength composition of light. [cf. Poynton, 2003, p 213]
18 cf. Poynton, 2003, p 211
19 cf. Selan, 2012, p 8
20 ARRI ALEXA Color Processing white paper, p 5

$$x = \frac{X}{X+Y+Z} \qquad y = \frac{Y}{X+Y+Z}$$

Equation 2.3: Normalizing the XYZ tristimulus values

The original X and Z tristimulus values can also be recovered by using the inverse of Equation 2.3[21]:

$$X = \frac{x}{y}Y \qquad Z = \frac{1-x-y}{y}Y$$

Equation 2.4: Recovering X and Z tristimulus values

A visual representation of the x and y coordinates is displayed in the ***CIE 1931 [x, y] chromaticity diagram*** which is generally used for showing the different primaries and color gamuts of color spaces.

On the edge (spectral locus) of the diagram are the spectral colors which are monochromatic and full saturated. The CIE 1931 [x, y] chromaticity diagram is an additive color model which means, the further it comes to the center, the color values became less saturation till they are getting white (white point). The good thing about the CIE [x, y] chromaticity diagram is its connection to the physical laws. The spectral locus represents the spectral colors with their wavelength in nanometers. E.g., the wavelength of orange-yellow is 580 nm but it could also be described by its tristimulus value (1.110, 0.787, 0) or a multiple of it which depends on the luminance used[22].

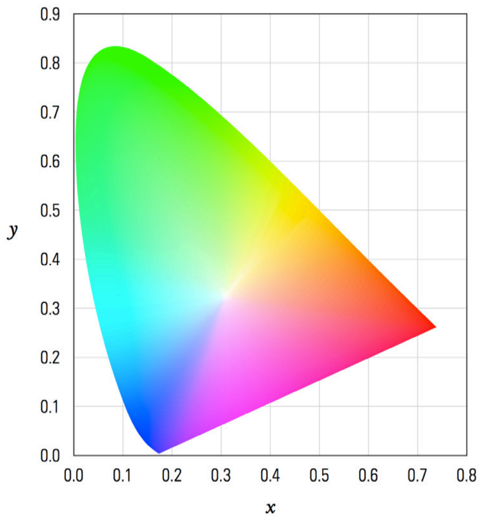


Figure 2.5: The CIE 1931 [x, y] Chromaticity Diagram

21 cf. Poynton, 2003, p 219
22 cf. Häßler, 2008

There are two more common forms in which the CIE space of visible color is expressed. These forms contain the same colors but distribute them differently. The *CIE L u'v' diagram*, developed in 1976, was created to get a better proportion in between the colors. It corrects the bad distortion of the CIE [x, y] chromaticity diagram and allows a much better visualization when comparing different color spaces. In the *CIE L*a*b diagram*, the visible colors are remapped and increased untill they fit conveniently into a square. This is a useful color space for editing purposes, mostly used by digital artist[23].

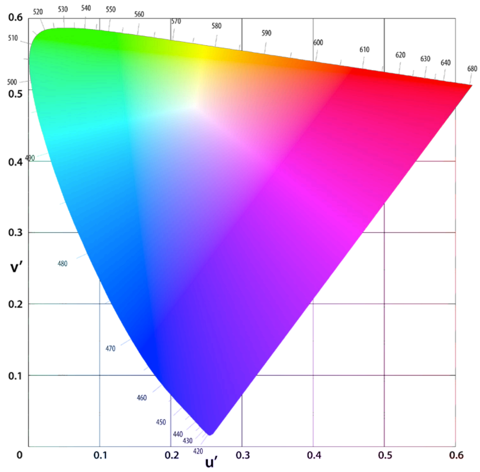


Figure 2.6: The CIE u'v' Diagram

2.3.2 Device-independent Spaces

The color space, standardized by the CIE in 1931, is used as a reference space. Nearly all software applications today are using this device-independent space as a reference color space in which each possible color of the human eye can be depicted[24]. Besides the device-independent color spaces, there are two more common types of color spaces which should be explained for a better understanding.

2.3.3 Device-dependent Spaces

Different from device-independent color spaces, a device-dependent color space needs a reference space to express the colors corresponding to them[25]. These

23 cf. McHugh, 2016
24 cf. McHugh, 2016
25 cf. McHugh, 2016

color spaces are limited to a specific range in the reference color space, e.g., the sRGB color space used by a monitor or the Adobe RGB 1998 color space used in a DSLR Camera.

2.3.4 Working Spaces

The third terminology in context of color spaces are the working spaces. These types of color spaces were used by applications to map the range of a device-dependent color space to the working color space.[26] This is a useful way to ensure that the software uses always the same range of color, which can be useful, e.g., for mathematical equations. A common working space could be sRGB IEC61966-2.1 in Adobe Photoshop or a linear color space in The Foundry's Nuke.

2.4 Basic Concepts of Color Transformations

"There are two basic methods of specifying a color in a three primary color system: by its three tristimulus values (T1, T2, T3), and by its chromaticity (t1, t2) and its luminance (Y). Given either one of these representations, it is possible to convert from one primary system to another."[27]

In a modern motion picture, the amount of different input capture devices is much higher than in former days. Especially in movies or series with different requirements on visual effect shots it is not unusual to use different cameras. What makes things worse is the increasing amount of different output devices such as cinema projectors, TV screens, smart phones or HDR monitors. Thus, a strong color management is more important than ever and the essential component of color management is converting or transforming colors from one space to another. When more different input/output devices are used, more color transformations have to be done and, as a result, this makes it more complex and chaotic. Therefore, it is important to know some basic designations and concepts of color transformations.

26 cf. McHugh, 2016
27 Pratt, 2001, p 709

2.4.1 Gamut Mapping / Color Space Conversion

A color transformation or color space conversion is described as the processing of components of an image within a specific color model (e.g., the CIE XYZ model). The conversion between the gamuts of two color spaces is referred as gamut mapping. As shown in *Figure 2.7*, the gamut of the Rec.709 primaries is smaller than the DCI-P3 gamut, which means, that an image with a Rec.709 color range will clip outside its gamut if it is displayed on a DCI-P3 output device. Thus, it has to be converted into that bigger DCI-P3 color space to prevent these issues. The same problem occurs on the reverse side because the bigger range of the DCI-P3 gamut has to be "compressed" when displaying on a Rec.709 monitor.

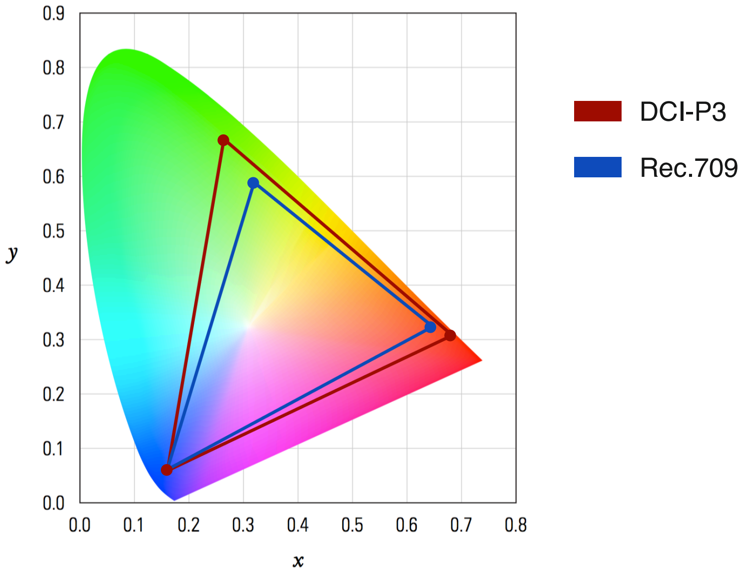


Figure 2.7: Rec.709 and DCI-P3 color gamuts inside the spectral locus of the CIE 1931 [x, y] Chromaticity Diagram

A color space conversion can be exercised in several ways. One option is using a Color Management Module (CMM) which is usually part of the used Color Management System (CMS). It uses different rendering intents (absolute and relative colorimetric, perceptual, and saturation) to prioritize quality appearances[28].

Another way of converting color spaces is the Color Transformation Language (CTL), developed by The Academy of Motion Picture Arts and Sciences' Science and Technology Council. It is a platform-independent interpreter and scripting language which *"has been designed for describing and implementing the operations that occur*

28 cf. McHugh, 2016

when images are moved from one color space to another during digital motion picture production."[29] [30]

2.4.2 Lookup Tables (LUTs)

As said, there are a few ways to achieve gamut mapping. The most common is using a lookup table (LUT). Jeremy Selan describes the main benefits of LUTs as followed: *"Lookup tables (LUTs) are a technique for optimizing the evaluation of functions [which] are expensive to compute and inexpensive to cache."*[31] Besides the significant performance gains, the lookup table has another big advantage because it can "bake" a series of different color transformations into one LUT which is often useful for distribution purposes and re-uses[32]. For example, a DI facility can "bake" their color conversions and corrections into one single LUT and send it to the VFX facility in one single LUT file. This process is explained more accurately in chapter 3.2.

2.4.2.1 1D LUTs

There are two types of lookup tables which are generally used in practice. The one-dimensional LUTs are indexed by a single variable. Thus, every basic color operation which needs just one variable can be assigned by a 1D LUT. This includes brightness, gamma and contrast operations. With a 1D LUT it is also possible to do color balancing operations, when a LUT is assigned individually to each color channel[33]. The advantage of a 1D LUT is its small size, fast calculation and the possibility to invert easily every color operation.

2.4.2.2 3D LUTs

The downside of 1D LUT is its limitation to a single variable which means that some color operations cannot be realized with it. For example, the "luminance operator"

29 Kainz, 2007, p 1

30 Note: The ACES implementation of OCIO is also defined by a set of CTL transforms which are baked via Python scripts into LUTs. [cf. Duiker, 2015, p 6]

31 Selan, 2012, p 45

32 cf. Selan, 2012, p 45

33 cf. Selan, 2012, p 45

needs more than one variable to convert a colored pixel into its grayscale equivalent. A 3D LUT solves this problem by allowing the use of three independent parameters. Within these additional variables, it is possible to do more complex color operations such as a hue and saturation modification. Because of its additional parameters, 3D LUTs can grow very quickly[34].

2.4.3 Color Decision List (CDL)

Every manufacturer has historically applied its color corrections in a different order. Thus, the "primary grade"[35] which is particularly used on a film set to do the first color corrections, loses its portability options. To bring back a bit of order, The American Society of Cinematographers (ASC) developed a color correction specification format called *Color Decision List (CDL)*. The CDL defines a strict order in which the color correction for a "primary grade" has to be realized[36]:

1. **Scaling / Slope (3 channels)**
2. **Offset (3 channels)**
3. **Power (exponent) (3 channels)**
4. **Saturation (scalar, with a fixed Rec.709 luminance target)**

Because of its simplicity, a CDL is often used as a "pre-grade" and can be delivered to the VFX facility, e.g., instead of a 3D LUT. Unfortunately, a CDL does not define a color space and should therefore be used with great caution to avoid any mistakes[37].

2.5 Scene-Referred Imagery versus Display-Referred Imagery

As mentioned earlier, color and brightness information can be encoded and saved in an image file in different ways. There are two types of image states. In

34 cf. Selan, 2012, p 46

35 Note: A "primary" grade is usually done onset to generate a first look that can be used for reviewing or proxy generation. It consists in most cases basic scaling and offset operations and sometimes a few saturation and gamma corrections. [cf. Selan, 2012, p 48]

36 cf. Selan, 2012, p 48

37 cf. Selan, 2012, p 48

the *"scene-referred imagery [are] the code values proportional to real world scene measurements"*[38]. That means the encoded image has its full range of colors – this is typically more than a monitor can display at once. Scene-referred images can be created through the characterization of a camera system[39]. In computer graphics, a scene-referred image state is also known as "scene linear" and is the usual way to render and save CG images. Scene-referred imagery is used as standard in visual effects because it can help a lot by integrating elements and CG renderings into the captured plate. The whole process of working in "scene linear" is described in more detail in chapter 3.2.2.

The second main image state is known as display – or output referred imagery. *"Display-referred imagery is defined colorimetrically with regards to an image as presented on a display."*[40] In a display-referred imagery the primaries of the color space of an image have to be inside the spectral locus and is limited in its range.

On post production, it is tried to stay in the scene-referred image state as long as possible to avoid losses in flexibility. But sometimes it happens, that images were rendered already in display-referred color spaces (e.g. consumer cameras). For working purposes, it is still possible to linearize such a display-referred image, e.g., an JPG file rendered in sRGB color space by doing a gamma correction.[41]

38 Selan, 2012, p 16
39 cf. Selan, 2012, p 16
40 Selan, 2012, p 16
41 Note: Gamma is described as a numerical value of the exponent of a power function which is applied to a monitor to obtain a linear perception. [cf. Poynton, 1998, p 4]

3 Color Management Workflow in a VFX Pipeline

Before a project starts, choosing the right workflow is essential for having a smooth production time without having problems which can't be fixed later on. Especially in producing visual effects, making the wrong decisions at the beginning of a show can rapidly increase time and costs.

The chapter starts with an overview of important parts of the whole post production pipeline while focusing on different approaches in a Digital Intermediate (DI) workflow. This is followed by a closer look at VFX pre-grades and exploring the best methods to manage their exchange between the DI and VFX companies. The next section is about preparing the plates for the visual effects work and what is important for an artist or producer in terms of color management. The chapter ends with an overview of the most common workflow problems and how to avoid them[42].

3.1 Workflow Basics in a Post Production Process

Nowadays, the post production of a motion picture is commonly separated in 4 main parts if VFX processes are included[43]:

- **Editing**
- **Visual Effects (VFX)**
- **Digital Intermediate (DI)**
- **Finishing**

In bigger movie productions, the tasks are usually separated from each other and will be processed in different companies that are specialized to it. Thus, it is necessary that the companies have a good communication among themselves.

After the movie is filmed, the footage is ingested and sent to the editor (also known as cutter) who brings the sequences and shots into the right order. This process takes often place with a close collaboration with the director. After the editing process is finished (picture lock), the editor usually sends his work via an Edit Decision List

---

42 Note: The given knowledge is based on my personal experience when working in VFX companies and learning from courses during my studies.

43 Note: Music and sound design is also a big part of the post production. It is not mentioned here because this paper is only treating the visual parts.

(EDL)[44] to the DI company. The Digital Intermediate process (DI) had its origin in the early 1990s. Because of a growing number of digital VFX shots, a new workflow was developed to create a digital form of the chemical film laboratory process. With that technique, it was possible to avoid the reuse of the film negative. Scanned once, the whole process was able to stay in the digital world which made the work on VFX shots much easier, especially in dealing with colors[45].

In the DI, the colorist imports the cut-list in its grading suite and creates the final look of the movie. Meanwhile the VFX company gets the raw plates from the production company or the DI company after the picture lock[46] and the VFX editor prepare the plates for the VFX shots, creates cut references and maintains the pre-grades which the DI department has created. After the visual effects are done, the final VFX shots are sent back to the DI company with the exact same color conditions as they got it from the DI. The last step is the finishing process (for the final distribution) which is also the job of the DI company. All the plates within the final grade, the credit list and the final sound, are brought together and rendered for the required output device (e.g., for cinema distribution a DCP format is necessary).

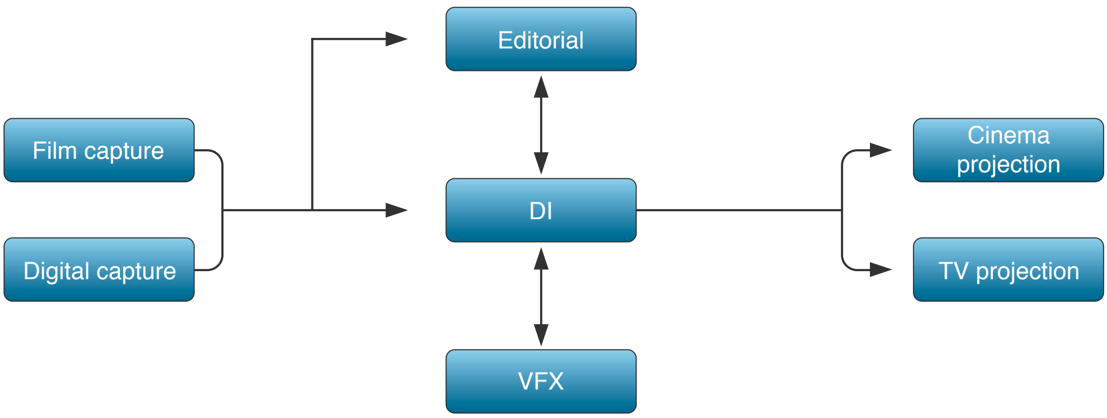


Figure 3.1: Post production workflow diagram

Sometimes these 4 processes are blending together, especially in smaller productions. It could happen that parts of the grading have to be done in a visual effects shot, e.g., a "day for night" shot[47] that needs more advanced rotoscoping work or a shot that has a complex retime.

44 "The [EDL] contains an ordered list of reel and timecode data representing where each video clip can be obtained in order to conform the final cut." [cf. <https://en.wikipedia.org/wiki/Edit_decision_list> (2016-07-29)]

45 cf. The Quantel Guide to Digital Intermediate, p 8-11

46 Note: In most cases the plates are delivered as DPX or EXR files.

47 Note: A "day for night" shot means, that a plate captured on daylight is graded to a night looking image.

3.1.1 Different Approaches for the Digital Intermediate Workflow

Grading in film or video got more complicated since digital cameras took over the market, because old workflows were still used and new workflows were developed. Two different approaches for a DI workflow should be explained in this thesis, because they are both still in use nowadays.

The first approach is called *"video-centric"*. It is based on the display-referred imagery and comes from the classical digital intermediate process in which the color corrections are applied directly to the output-device color space. Thus, no viewing transformation is necessary, which has its advantages because of much simpler color processes and less risk of producing artefacts[48]. The big disadvantage of this approach is obviously the limitation of colors when grading in a limited color space.

Figure 3.2: The „video-centric" DI workflow

The second *"film-centric"* approach is based on the scene-referred image state. The color corrections are directly applied in a working color space (device-independent) but are viewed through a viewing transform (e.g. a 3D LUT)[49]. With this technique, it is possible to use the whole color range and achieve much more precision. It builds also the fundamentals of the ACES workflow which is described in chapter 5.

Figure 3.3: The „film-centric" DI workflow

Another common approach of the film-centric workflow is the *"Log-Grading"* in which the colorist does its color corrections directly in a log space instead of first converting in a scene-linear or similar color space and transforming afterwards into the output-device color space. The reason for that approach has its origin in the classic film grading with the "Kodak Cineon logarithmic encoding" and its similarity with the log spaces of digital cameras nowadays[50]. Because of different "feelings" and behaviors of a color correction in a linear color space (e.g.), in comparison to

48 cf. Selan, 2012, p 27
49 cf. Selan, 2012, p 27
50 cf. The Quantel Guideto Digital Intermediate, p 52

a logarithmic color encoding, using the "Log-Grading" is still common. *Figure 3.4* is showing an example in which a scene-linear rendered image is converted into a logarithmic space and transformed into sRGB color space after the grade.

Figure 3.4a: Log-Grading example - scene-linear rendered image

Figure 3.4b: Log-Grading example - log converted image for DI

Figure 3.4c: Log-Grading example - final sRGB image after grade

3.2 VFX Pre-Grade

In a movie production with a high amount of visual effects shots, the DI facility usually creates pre-grades of these shots and send them to the VFX facility. The pre-grades serve as reference for the VFX artists to match the digital effects to the wishes of the director or cinematographer. For example, in a "day for night" shot, a pre-grade is done to simulate a night look. This can help an artist to a greater focus on the important parts of the image. A VFX pre-grade could be also made to compensate light changes, e.g., when shooting a sequence on several days with different light conditions. Thus, the artists can review their work with the same look for all shots of this sequence.

As mentioned in chapter 2.4.2, working with lookup tables when creating VFX pre-grades is widely used because of its performance benefits[51]. Thus, a VFX pre-grade is also designated as *"preview LUT"*. Another common way to work with is a CDL (chapter 2.4.3) which is also provided by the DI facility.

3.2.1 Cooperation with the DI Facility

In a motion picture production, the DI facility usually creates the preview LUTs for the VFX facility and exchanges them. The VFX editor receives the LUTs and has to adjust the original plates to the correct look. The conversion of the color spaces before grading the image are done independently between the DI- and VFX facilities. Therefore, the VFX editor should exactly know the workflow processes in which the DI facility did the color conversions and corrections. If the processes are well documented and shared by the DI facility, the VFX editor is able to prevent color miss matches when bringing the preview LUTs into the internal color pipeline, because the order of the different operations are established. A good exchange can also include a reference image which is provided by the DI facility to compare the matched plates. A well-defined pipeline document has generally listed the following information in terms of the color workflow:

- **The correct color working space**
- **A description of the given LUTs**
- **The correct order of operations for matching the dailies**
- **Some notes on possible errors that may occur**

51 cf. Selan, 2012, p 46

3.2.2 Preparing the Plates

Before the Artists can start with their work on a shot, the VFX editor has to prepare the incoming footage for injection into the internal VFX pipeline. Color management is a very important part of this process, because the original footage has to be converted into the correct working color space.

In computer graphics, most software applications are calculating their mathematical operations in linear light space considering the behaviors of physics in the real world. The human eye is much more sensitive to lower parts of the light. Thus, calculating in a linear mode means the result is more weighted in areas with brighter source colors than in darker areas[52].

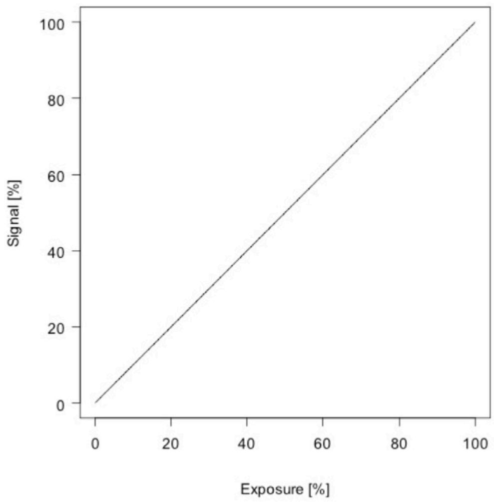


Figure 3.5a: Tone characteristic of Linear encoded image

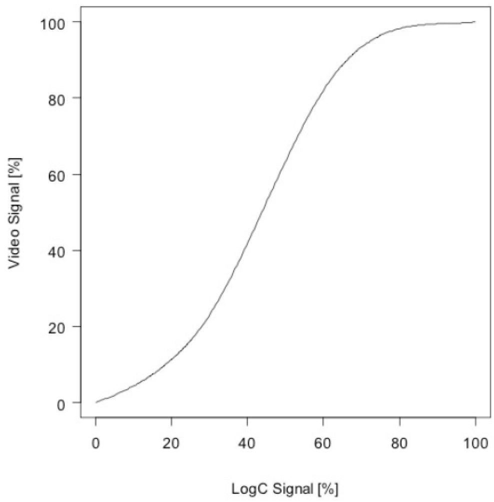


Figure 3.5b: Tone characteristic of LogC encoded image (including gamma correction)

In most cases, the VFX editor converts the incoming footage into scene-linear color space. For example, if the footage has captured with an ARRI Alexa within logC encoded images, the VFX editor has to undo the logC curve to obtain linear data[53]. After its color space conversion, the images are generally saved in the OpenEXR 16-bit (half) float image format[54]. OpenEXR is the standard format when working in linear color space. Because of its fast performance and many useful features, such

52 cf. Meyer, p1

53 cf. ARRI ALEXA Color Processing white paper, 2011, p 12

54 "OpenEXR is an open-source image format creating by ILM in 1999, which has near universal adoption in the VFX and animation industries. EXR is primarily intended for storing floating point, scene- referred imagery." [Selan, 2012, p 49]

as supporting multi-part image files, storing up to 32-bit floating point images or the support of deep data, most VFX companies are using it over the entire production[55].

The next step in the VFX production is setting up the color management, which means in most VFX facilities setting up the OCIO configuration (chapter 4.3) and bringing the received LUTs and the created plates into the right folder structure.

3.3 Workflow Problems – common Color Management Issues

Because of the complexity when dealing with colors in the digital world, there is always a high potential to do something wrong. Sometimes it is just a simple color transformation which is not applied, or a wrong named lookup table causes wrong colors at the end. Furthermore, it can happen that some errors are noticed just in the end of the process, e.g., when previewing the final shots on another output device. One of the biggest problems occurs, when the exchange process is not clarified good enough and the VFX editor has to guess what the DI facility did. In the worst case, there is no documentation and given preview LUT to work with.

55 cf. <http://www.openexr.com/> (2016-08-01)

4 OpenColorIO (OCIO)

As described in the previews chapters, there are a lot different color spaces and transformations in the post production process. It starts and ends with the DI process in which the footage is transformed in the working color space, graded and transformed again for the final output device. Between the DI processes stands the visual effects production, which also needs to handle a lot of color transformations for converting the color space into scene linear or preparing all the VFX pre-grades.

To keep track of all the color spaces and LUTs, a strong color management solution is essential. One of the most used color management solutions, especially when it comes to visual effects, is OpenColorIO. The intention of this chapter is to establish an overview about this software. Beginning with a description of the potential applications of the OCIO workflow, the chapter gives an introduction on how to setup an OCIO configuration[56]. In the last part of this chapter, I will take a brief look at the OCIO toolset and how to use it.

4.1 What is OpenColorIO?

"OpenColorIO (OCIO) is a complete color management solution geared towards motion picture production with an emphasis on visual effects and computer animation."[57] OCIO is supported by many software applications like The Foundry's Nuke, Blender, Mari or RV and its core API is available for use in C++ and Python[58]. It has been developed since 2003 by Jeremy Selan[59] and released at version 1.0[60], sponsored by Sony Pictures Imageworks (SPI)[61]. OCIO is also compatible with the Academy Color Encoding Specification (ACES) and supports many popular LUT-formats on all platforms. It is also designed to handle color workflows in both scene-referred and display-referred imagery[62].

Within the OCIO configuration it is possible to set as many color transformations or look definitions as needed in one single config file. Thus, the big advantage of

56 Note: a practical example is given in chapter 6.3.2.
57 <http://opencolorio.org> (2016-08-01)
58 cf. <http://opencolorio.org/CompatibleSoftware.html> (2016-08-01)
59 cf. Selan, 2012, p 43
60 Current version 1.0.8
61 cf. <http://opensource.imageworks.com/> (2016-08-01)
62 cf. Selan, 2012, p 43

OCIO is the possibility to manage the color pipeline from a single location. Once the folder structure and the OCIO configuration is set, all supported applications are able to access and use this configuration and by changing it all, applications will apply the changes. For example, if a new shot is added for an existing show, the new preview LUT just has to be implemented into the OCIO folder structure and can be used immediately by the compositors, CG artists or supervisors. Another advantage of this workflow is the limited amount of people who are allowed to change the OCIO settings, e.g., if all artists would manage the color workflow for their shots by themselves, the risk that someone is doing something wrong is much higher than if the changes are done by one or two selected artists[63].

4.2 Potential Applications in VFX Processes

The daily work of a VFX artist or supervisor can be much easier and faster if a well-set OCIO configuration is used. A few potential applications in which OpenColorIO can improve the workflow will be introduced in the following.

4.2.1 Compositing

Compositing is the final process in a creative aspect of a VFX production. Steve Wright describes the artistic side of compositing as followed: *"The ultimate artistic objective of a digital composite is to take images from a variety of different sources and combine them in such a way that they appear to have been shot at the same time, under the same lighting conditions, with the same camera."*[64] The technical meaning of compositing could be described as different mathematical operations which are applied to images in a linear color space.

Compositing is the most important part in terms of color management during visual effects, because several input images, with maybe different color tones or color spaces, are composed together and has to be fit perfectly as a resulting new image. For example, a complex compositing task could contain CG renderings, matte paintings, different plates or additional stock footage such as smoke and fire. To

63 Note: In most facilities the OCIO config is created and managed by the VFX editor, the heads of or the supervisors.

64 Wright, 2010, p 1

bring all these elements together, most compositing applications are converting the incoming images into a linear color space. By calculating the images in the same linear work space, it is ensured that no mathematical disparities will appear.

The industry standard application for compositing is The Foundry's Nuke[65]. Inside the application, every loaded image is automatically converted from its original color space into a linear light space. To ensure the correct color space conversion, Nuke interprets the metadata from the incoming image to determine its actual color space. For example, if a JPG image, which is encoded in sRGB color space is loaded, Nuke does automatically transform the sRGB curve into scene linear.

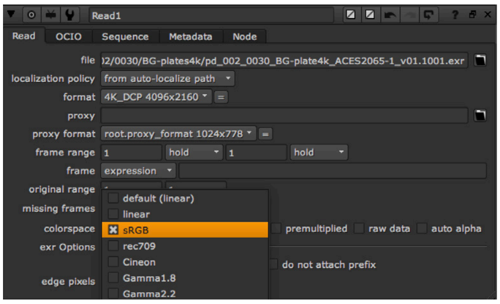


Figure 4.1: Color Space selection in Nuke's Read node

Because Nuke is working internally in linear light space, the linearized image looks a way too dark on the viewing device[66]. Thus, Nuke is providing the user with an option for choosing a viewer LUT which is basically the same as a preview LUT (see chapter 3.2). The viewer LUT does not affect the image, it just applies a gamma correction to the selected viewer.

Nuke has a well-designed implementation of the OCIO workflow including options for loading the OCIO config and working with some useful OCIO nodes which is described in more detail later in this paper. Furthermore, Technical Directors (TDs) have the possibility to develop their own tools because of their availability for use in python which is a powerful scripting engine in Nuke[67].

65 cf. <http://www.thefoundry.co.uk> (2016-07-20)

66 Note: An example of a linearized image is shown in Figure 3.4a.

67 cf. <https://www.thefoundry.co.uk/support/developers/> (2016-08-02)

4.2.2 CGI/FX

The creation of Computer Generated Images (CGI) and smoke or fire elements (FX) is an essential part of a visual effects company. Because CG images are created from scratch and generally rendered in a scene linear space there is no color conversion necessary if it ends up in compositing. Though using a color management tool like OCIO can be useful anyway. For example, when looking at a linear rendering, the CG artist needs a gamma corrected output in form of a preview LUT as well as the compositor. Fortunately, strong render engines like Chaosgroup's V-Ray[68] or Solid Angle's Arnold[69] are still supporting OCIO for display transformations (e.g., on Maya's frame buffer). Furthermore, within V-Ray it is even possible to use the OCIO config in a specific "VRayOCIO" texture managing the color spaces and looks of textures directly in the CG application[70].

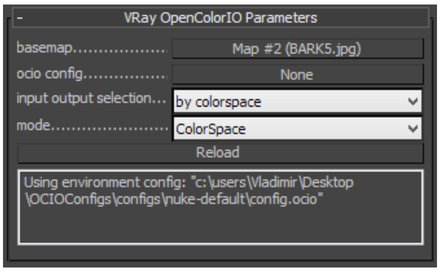


Figure 4.2: V-Ray OCIO support in Maya

4.2.3 Matte Painting

If the camera movement in Z-direction (into depth) of a VFX shot is not too heavy or a faraway background needs to be replaced, it is a usual method doing that without a CG render and using instead a digital matte painting. A matte painting is described as a two-dimensional painting consisting of one or more images. Back in compositing, the matte painting is projected on a geometry (Projection Mapping) and rendered through the match moved camera or is tracked directly on the camera movement.

The standard application for doing matte paintings in a visual effects facility is Adobe's Photoshop[71] which is a powerful painting tool but unfortunately has some weaknesses in terms of color management and the floating point workflow, because

68 cf. <https://docs.chaosgroup.com/display/VRAY3MAX/OpenColorIO+Support> (2016-08-02)
69 cf. <https://support.solidangle.com/display/AFMUG/1.2.7.0> (2016-08-02)
70 cf. <https://docs.chaosgroup.com/display/VRAY3MAX/OCIO+Map+|+VRayOCIO> (2016-08-02)
71 cf. <http://www.adobe.com/de/products/photoshop.html> (2016-08-02)

it is more focused on print and graphic-design than supporting modern feature-film approaches[72].

The first problem comes when working with a linear floating point image that is provided by the compositing or DI department (e.g. a 16-bit floating point EXR file). Although it is possible to work with floating point images, it is not recommended because some important tools in Photoshop are only usable within a 16-bit integer image workflow.

The second big problem when working with Photoshop are the limited color management options, especially for VFX requirements. Besides the fact that the current version of Photoshop is not supporting OpenColorIO, there are no real possibilities to load 1D or 3D LUTs into the application[73]. Generally, there are three main approaches of using Photoshop in a VFX pipeline anyway.

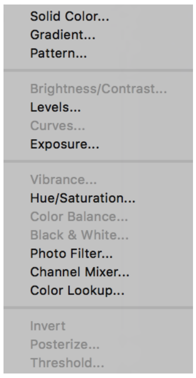


Figure 4.3: Limited Photoshop Tools
Working with a 16-bit linear floating point image in Photoshop is causing a limitation of some important color correction tools and filters (greyed out)

The first approach is to work with linear floating point data which means to lose some important tools that would be enabled when working with 8 or 16-bit integer images[74].

The second approach is to do the matte painting in log space to achieve that the image is converted with a log-function and saved as an 8 or 16-bit integer image[75]. After the matte painting is done, the image needs the reversed operation in form of a log-to-linear conversion. With that approach, the painting process is more precisely because of the flat log image which retains the dark and bright levels. Besides, it is also possible to use all tools when working with an 8 or 16-bit integer image. The big disadvantage of painting with log space in Photoshop is the clipping of all values above 1 and under 0. Nevertheless, it is a very common approach when doing matte paintings in a VFX facility.

The third approach is to work with a created ICC profile which simulates the graded look of an image. To achieve this workflow, the original image is mapped first

72 Note: Photoshop version CC (2015.5) at the time this paper is written.
73 Note: The color management in Photoshop is based on working with ICC profiles.
74 cf. Wheatley, 2016, p 23
75 cf. Wheatley, 2016, p 23

into a [0.0, 1.0] range and is saved as an 8- or 16-bit integer file. The next step is creating a ICC profile, for example by using the OpenColorIO tools and import them into the right direction of Photoshop[76]. A detailed example of this straight forward technique is given in chapter 6.5.1.

4.2.4 Reviewing in RV

Reviewing the work of a VFX shot with the preview LUT or any other useful gamma curve is essential for a comp or visual effects supervisor. One of the most popular playback tool in the industry is RV published by tweaksoftware[77]. Since the release of RV 4, OCIO has been implemented in the software package. Because of some limitations when working with an OCIO configuration in RV, VFX facilities often develop their own OCIO implementation to adapt the tool to their pipeline. For example, ***Pixomondo***[78] has written its own implementation in order to have the possibility to load different LUTs for each shot that is viewed in the same RV session. This is not supported in the standard implementation because only one show LUT can be loaded no matter how many shots are viewed at the same time in RV[79].

4.3 Workflow Basics – setting up a OCIO Configuration

A basic setup must be created before starting to work with OpenColorIO. This chapter is exploring some fundamental concepts of the OCIO configuration and the most important functions that will be declared in the OCIO configuration file.

4.3.1 Install OpenColorIO

At the time this thesis is written, the OCIO core library is only available at source code[80] which means prebuilt binaries are not yet available for all platforms and has

76 cf. Wheatley, 2016, p 23
77 cf. <http://www.tweaksoftware.com/products> (2016-08-02)
78 cf. <http://www.pixomondo.com/> (2016-08-21)
79 cf. <http://tweaksoftware.com/static/documentation/rv/current/html/opencolorio_integration.html> (2016-08-02)
80 Version 1.0.8

to be compiled by the user himself or using several package managers such as Homebrew for OS X or RedHat Enterprise (RHEL) for Linux[81].

I installed the software on an OS X platform via the Homebrew package manager which was an easy step by only typing a command line into the terminal.

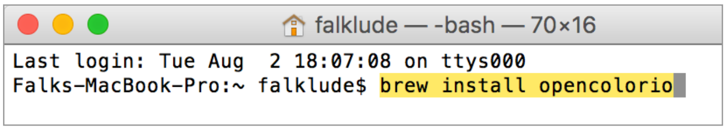


Figure 4.4: Install OCIO via Homebrew

After installing the software, I was able to run the OCIO tools (e.g., ociocheck or ociobakelut) by executing the commands in the terminal, which was quite helpful when working the first time with an OCIO configuration.

The most efficient way to setup an OpeColorIO configuration is using one of the already existing configuration packages which are provided for free by Sony Pictures Imageworks[82]. By downloading the .zip file from the website, there are a couple of predefined OCIO configuration setups, each including separate config.ocio files and folders with baked LUTs or other color transformation setups. Thus, it is possible to use one of the setups as a start configuration and edit it as needed. The three most useful setups are the "spi-vfx" when setting up a standard VFX pipeline, the "spi-anim" setup for working in a CGI-only project and the "aces_1.0.1" config, which is the basic setup for the practical example in chapter 6.

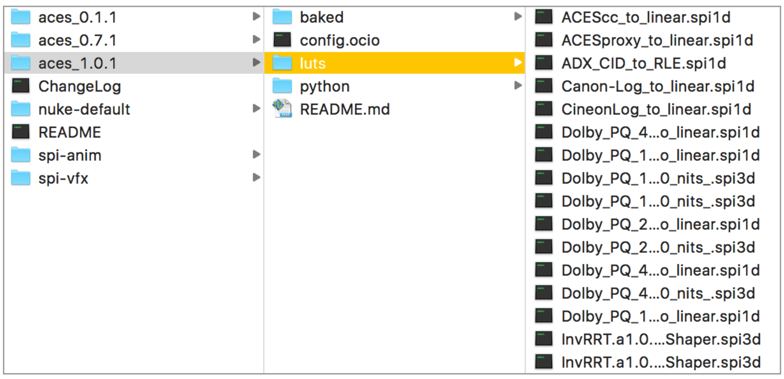


Figure 4.5: OCIO Start Configurations

81 Note: Unfortunately, there is still no prebuild version actually available for a Windows platform (August, 2016).
82 cf. <http://opencolorio.org/configurations/index.html> (2016-08-02)

4.3.2 OCIO Configuration Syntax

The folder structure and organisation of an OCIO configuration is primarily controlled by a text file, usually named “config.ocio”. Within this file there are some basic statements and some required and optional commands. The following describes the most important required commands of the configuration syntax as well as some useful optional ones[83].

4.3.2.1 Required Commands

ocio_profile_version The configuration always starts with the indication of the used profile version. At the time this paper is writtenn there exists only one version of the OCIO profile, so its value can only be 1 (one)[84].

```
ocio_profile_version: 1
```

roles *“A “role” is an alias to a colorspace, which can be used by applications to perform task-specific color transforms without requiring the user to select a colorspace by name.”*[85] For example, the “OCIOLogConvert” node in Nuke works with roles which prevent an artist by doing something wrong. It is also possible to create own roles if the given role name is defined by the required application or part of the filename itself.

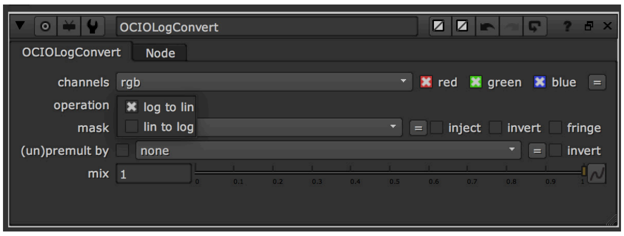


Figure 4.6: OCIOLogConvert

```
roles:
  color_picking: sRGB
  color_timing: AlexaV3LogC
  compositing_log: AlexaV3LogC
  data: raw
```

83 Note: A more detailed explanation with examples of the OCIO syntax is written in the online documentation.
84 cf. <http://opencolorio.org/userguide/config_syntax.html> (2016-08-03)
85 <http://opencolorio.org/userguide/config_syntax.html> (2016-08-03)

displays This section gives the user the possibility to define the color treatment on different display devices (e.g. a DCI-P3 grading monitor or a sRGB display device). It is possible to give each of the display devices different "views" which allow different ways to interpret an image on the display device[86]. For example, a preview LUT used in Nuke can be created with the "views" option in the OCIO configuration.

```
displays:
  default:
    - !<View> {name: Dailies, colorspace: AlexaV3LogC, looks: dailies}
    - !<View> {name: Alexa, colorspace: AlexaV3LogC, looks: alexa-3d}
    - !<View> {name: sRGB, colorspace: sRGB}
    - !<View> {name: None, colorspace: raw}

active_displays: [default] (optional)
active_views: [Dailies] (optional)
```

colorspaces This is the main section of the OCIO configuration file. Color spaces can be defined and used in all supported applications as well as in the config. ocio itself when setting up "looks" (chapter 4.3.2.2) or display views as mentioned above. The colorspace command can specify different types like a name, the bitdepth (optional), a description (optional) or family (optional), to mention some of them. The main part of the colorspace command is required to do color transformations. There are a few possibilities to do that in OCIO.

For example, one option is operating with a reference color space, which means that *"all other color spaces are defined as transforms either to or from this [reference] color space."*[87] This is a powerful concept by first converting all images to a reference color space and then apply all further color transformations from that same color space. Thus, it is much easier to organize the color pipeline and combining the different LUTs when using multiple transforms in a OCIO colorspace.

Color space transforms can be achieved either by using the "ColorSpace Transform" command which is doing an internal color space conversion from a specified source space to a destination space, or by using the "FileTransform" command which uses an LUT file directly for the transformation. The "ColorSpaceTransform" and "FileTransform" are the most used commands to specify a color space in OCIO, but there are a few other useful transforms available, e.g. "CDLTransform", "ExponentTransform" (gamma correction), "GroupTransform", "MatrixTransform ", "AllocationTransform", "LogTransform" and "LookTransform", to mention a few.

86 cf. <http://opencolorio.org/userguide/config_syntax.html> (2016-08-03)
87 <http://opencolorio.org/userguide/config_syntax.html> (2016-08-03)

```
!<ColorSpace>
  name: srgb8
  bitdepth: 8ui
  description: |
    srgb8 :rgb display space for the srgb standard.
  from_reference: !<GroupTransform>
    children:
      - !<ColorSpaceTransform> {src: lnf, dst: lg16}
      - !<FileTransform> {src: lg16_to_srgb8.spi3d, interpolation: linear}
```

4.3.2.2 Optional Commands

search_path The search_path section is specifying the list of directories where the files (e.g. a LUT) can be localized. This can become a very powerful command, especially when working on a big show with a lot of different preview LUTs. For example, when having a folder structure with separate sequence and shot directories, it is possible to specify them by setting up different types of variables.

```
search_path: luts:luts\${SEQ}\${SHOT}:luts\_main
```

looks *"A "look" is a named color transform, intended to modify the look of an image in a "creative" manner (as opposed to a colorspace definition which tends to be technically/mathematically defined)."*[88] It can be described as an on-top color correction which is similar to the OCIO colorspace, but differs by specifying a "process space" in which the color transform is applied. Thus, a "look" is generally used to apply the preview LUT file which came from the DI facility and use it in a display view setup.

```
looks:
  - !<Look>
    name: previewLUT
    process_space: AlexaV3LogC
    transform: !<GroupTransform>
      children:
        - !<FileTransform> {src: PreGradeLUT.cube, interpolation: linear}

displays:
  default:
    - !<View> {name: Dailies, colorspace: AlexaV3LogC, looks: previewLUT}
```

88 <http://opencolorio.org/userguide/looks.html#userguide-looks> (2016-08-04)

equalitygroup By specifying an "equalitygroup" in the "colorspace" section, OCIO is able to identify non-operation transforms, e.g., transforms between two identical color spaces which just differ from their bit-depths and don't require a transform operation. By avoiding those unnecessary transform steps the process time can become (much) faster.

family the "family" section is an useful option to organize color spaces within an UI by putting them in different groups. Unlike the "equalitygroup", it is not affecting any transform operations.

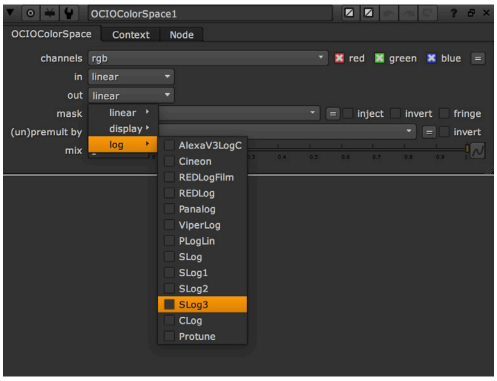


Figure 4.7: Family Groups in Nuke

4.3.3 YAML Format

The config.ocio file is based on a YAML format which *"takes concepts from programming languages such as C, Perl, and Python, and ideas from XML and the data format of electronic mail (RFC 2822)."*[89] It could be worth to take a look first at the basic syntax before starting to work with an OCIO configuration.

89 <https://en.wikipedia.org/wiki/YAML> (2016-08-04)

5 The ACES Workflow

As discovered in the previous chapter, the OCIO software package is helping a lot to manage the color pipeline in a visual effects production. But it is just a help for managing the “chaos” which is coming from the different approaches when dealing with colors. To bring order into the whole post production process, the Science and Technology Council of the Academy of Motion Picture Arts and Sciences (AMPAS) defined a color encoding scheme which can describe all (and even more) possible colors of the CIE color spectrum and full dynamic range imagery. It is called “Academy Color Encoding Specification” (ACES) and is described in more detail in this chapter.

After giving a quick overview about the ACES project, I will describe the basic workflow steps. The chapter continues with a short introduction of the EXR file format in which ACES files are stored. An overview of the advantages and disadvantages of the ACES workflow is given in the last part of the chapter.

5.1 What is the ACES Project?

The Academy Color Encoding System (ACES) is a free, device independent image interchanged system which is specified by the Society of Motion Picture and Television Engineers (SMPTE) in 2012[90]. It was part of many motion pictures such as *Oblivion* (2013), *Elysium* (2013) or *Legend of Tarzan* (2016), to name some of them[91]. Since its release of version 1.0, ACES is strongly integrated in the most common post production software tools as well as in the official OCIO-Configs repository.

The Academy describes ACES as followed: *“[ACES] can be applied to almost any current or future workflow. It was developed by hundreds of the industry’s top scientists, engineers and end users, working together under the auspices of the Academy of Motion Picture Arts and Sciences.”*[92]

In a summary, the ACES workflow is a “film-centric” color approach defined by a scene-referred working space and a reference viewing transform. By using ACES within its specified color encodings, it enables a full dynamic range and full color space post production workflow. Thus, DI and VFX facilities are using the same

---

90 SMPTE ST 2065-1:2012

91 cf. <http://www.imdb.com/search/title?colors=aces> (2016-08-04)

92 ACES Documentation Guide, 2016, p 6

working space, which means a great benefit especially for the VFX facility, because they don't have to struggle with transform issues, e.g., when applying the LUTs from the DI facility.

5.2 Workflow Basics

The ACES color space is actually a part of the Image Interchange Workflow (IIF) which covers all aspects of the post production process, beginning from capturing the image, over grading, to final output. In terms of color management, the IIF workflow is generally referred to as the ACES workflow.

The ACES workflow consists of 4 sections which are described in more detail following this chapter. In the first step, the recorded images are converted into the ACES color space with the Input Device Transform (IDT)[93]. Next, the ACES encoded scene linear images are converted in an output-referred image state. This happens with the Reference Rendering Transform (RRT), which converts the images into the Output Color Encoding Space (OCES). In the last step the images are transformed into the color space of the desired output-device, e.g. a Rec.709 HD monitor. This correction happens in the Output Device Transform (ODT).

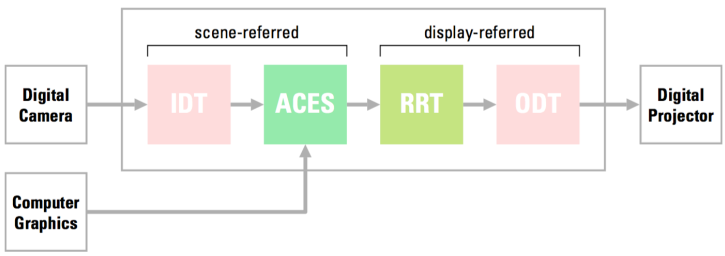


Figure 5.1: Top-level ACES workflow

5.2.1 Input Device Transform (IDT)

The color mapping into the ACES color space is the first step of the ACES workflow. In the ACES documentation the IDT is described as followed: *"In the Academy Color Encoding System, an Input Device Transform (IDT) processes non-color-rendered*

93 Note: It is not a mandatory requirement to convert into an ACES defined color space when working with the ACES workflow. Other color spaces could also be used, e.g., a log space. Though, to exploit the full scope of possibilities, it is recommended to use the ACES color space.

RGB image values from a digital camera system's capture of a scene lit by an assumed illumination source (the scene adopted white). The results of this process are white-balanced ACES RGB relative exposure values."[94]

IDTs are generally designed, individual for each camera model, by the camera manufacturers, because they have the best understandings of their devices. The Academy is recommending two sets of IDTs for each camera model, one optimized for daylight (CIE Illuminant D55), and one optimized for tungsten (ISO 7589 Studio Tungsten)[95].

To ensure that each IDT is producing the correct ACES RGB relative exposer values, the Academy developed a hypothetical camera device called ***"Reference Input Camera Device"*** (RICD) to give the manufacturers a matching reference to work with. The RICD is a *"[...] colorimetric camera whose spectral sensitivities are exactly expressed as linear transformations of the CIE 1931 colorimetric observer."*[96]

5.2.2 The ACES Color Encoding

The ACES color space is defined as the standard working space in the IIF workflow. Within the first release, the ACES primaries[97] were designed to enclose the entire spectrum of visible light (see ***Figure 5.1***). This color encoding was specified as ***"ACES2065-1"*** by the Science and Technology Council of the Academy of Motion Picture Arts and Sciences (AMPAS).

5.2.2.1 ACES2065-1

ACES2065-1 is the fundamental colorimetric encoding in the ACES framework and is often used *"[...] for exchange [or archiving] of scene-referred linear image data, between and throughout production and postproduction"*[98]. Because of causing some problems when working with the ACES2065-1 color space, the Academy developed additional types of encodings for different tasks in the post production.

94 Procedure P-2013-001, 2016, p 7
95 cf. Procedure P-2013-001, 2016, p 7
96 Procedure P-2013-001, 2016, p 10
97 AP0 primaries
98 cf. Specification S-2014-004, 2016, p 6

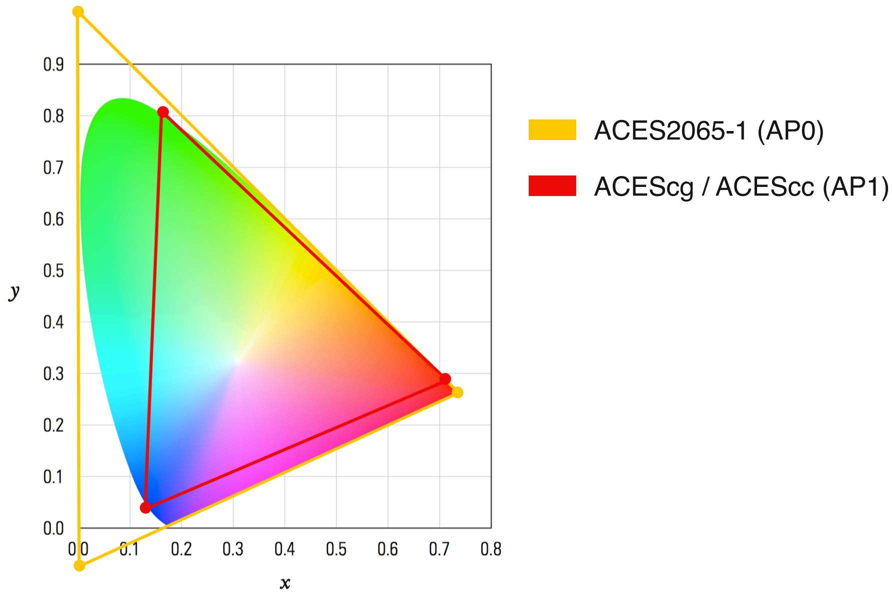


Figure 5.2: The ACES color encodings (AP0 and AP1 primaries)

5.2.2.2 ACEScc / ACESproxy

ACEScc is described as a logarithm encoding of ACES with primaries[99] similar to grading devices (e.g. DCI-P3). It is basically developed for using in DI and not intended for interchange or archiving. There is another encoding specification for use in 10-bit and 12-bit integer systems called ACESproxy. It has the same AP1 color primaries as the ACEScc color encoding but only a restricted range of values which correspond to a range between 0.0 and 1.0 of the ACEScc encoding[100].

5.2.2.3 ACEScg

Using the ACES2065-1 encoding, within its very wide-gamut primaries, for CGI productions can cause a lot of issues which are described by Kevin Wheatley, the Head of Imaging at Framestore[101], as followed: *"When using the [ACES2065-1 encoded image files] as-is inside our CG rendering, they produced undesirable effects caused by multiple light bounces tending towards the primaries, resulting in unwanted hue [behaviors] when transforming back to real image gamut."*[102]

99 AP1 primaries
100 cf. Specification S-2014-003, 2016, p 5
101 cf. <https://www.framestore.com/> (2016-08-21)
102 Wheatley, 2016, p 20

To avoid these issues, the Academy has developed the ACEScg encoding specification which is reasonably the same as ACES2065-1, but uses the same AP1 color primaries as the ACEScc working space[103].

To convert between ACES2065-1 and ACEScg, a simple transformation matrix is applied. Calculating this function is also possible within the OCIO configuration by using the “MatrixTransform” command[104].

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

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

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

Figure 5.3: Transformation Matrix ACES to ACEScg

ACEScg is a powerful working space for visual effects processes and already integrated in color management pipelines. For example, Framestore has released a paper at the DigiPro 2016 conference in which Kevin Wheatley, the Head of Imaging at Framestore, describes how they integrated ACEScg as standard color space in their color pipeline[105].

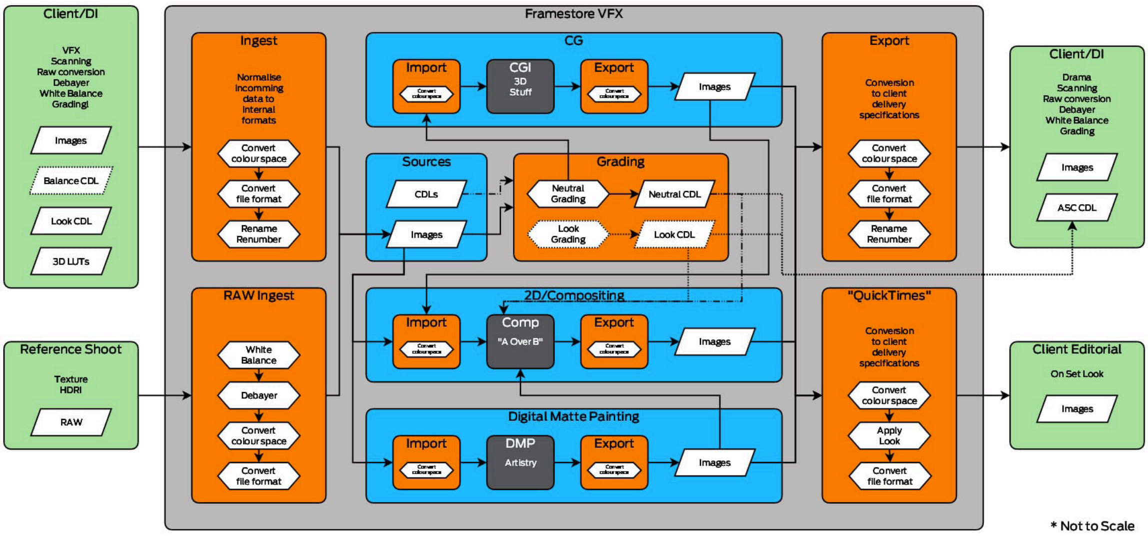


Figure 5.4: The simplified form of the VFX pipeline at Framestore with focus on color management

103 cf. Specification S-2014-004, 2016, p 6

104 cf. Specification S-2014-004, 2016, p 8

105 cf. <http://dp2016.digiproconf.org> (2016-08-05)

5.2.3 Reference Rendering Transform (RRT)

A scene-referred, linear ACES image file cannot be properly assessed if viewed on a grading display device. Because of extended dynamic range values, the image is looking too dark and unnatural for the human eye. Therefore, it has to be converted into an output-referred color space of a typical output device. The conversion is taking place in the Reference Rendering Transform (RRT).

The relation between RRT and an ACES image is similar to the classical print film to the camera negative. In other words, the ACES image could be seen as a digital negative and the RRT as the print stock[106]. One of the major goals by designing the RRT was to simulate the typical "film look" which is caused on its light reaction when passing through different layers of the film material[107].

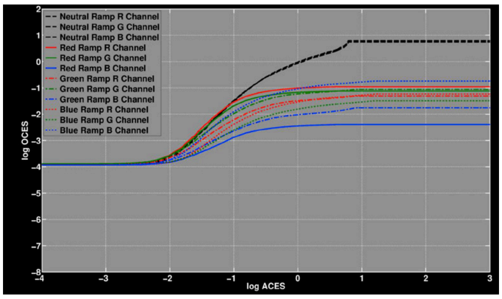


Figure 5.5a: Print Film emulation

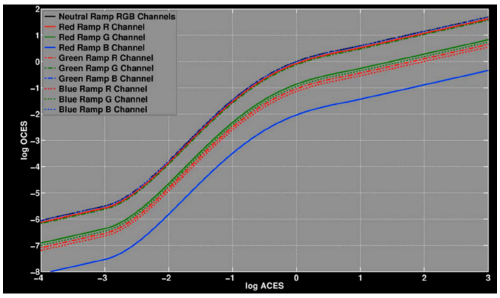


Figure 5.5b: RRT curve (release candidate, Feb 2012)

The RRT is mapping the ACES color representation to the Output Color Encoding Space (OCES), which has the same chromaticity primaries as ACES, a linear encoding and a similar large contrast range[108].

5.2.4 Output Rendering Transform (ODT)

To view RRT generated OCES images on the current output device, they have to be combined with an Output Rendering Transform (ODT). The ODT is the last transformation step of the IIF/ACES workflow. *"It maps the high dynamic range, wide color gamut OCES RGB values to a smaller range and color space that is appropriate*

106 cf. Shaw, 2012, p 11

107 Note: The RRT is not creating a "film look", it is just a carefully calculated starting point for color grading. [cf. Shaw, 2012, p 12]

108 cf. Tooms, 2015, p 586

for the display, so that the colors are always the same to the eye.”[109] In more technical speaking, the ODT applies a simple tone curve, that reduces the dynamic range of the OCES image to the dynamic range of the required output device. To ensure that every output device has its right color space conversion, many different ODTs are developed.

The great thing about the RRT-ODT workflow is the possibility of a future proofed storage. For example, a movie could be archived in ACES color space and future changes could be made easily by applying a better and contemporary RRT or an ODT, which is made for a possible new output device.

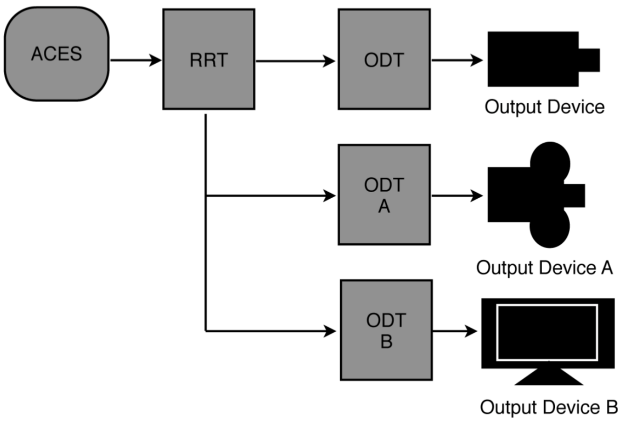


Figure 5.6: The RRT/ODT workflow

5.2.5 Look Modification Transform (LMT)

In the ACES workflow the grading step is done on ACES encoded images, between IDT and RRT/ODT. Though, in some cases it can be useful to apply another “fixed” color transform before the RRT and after the actual grade. This section is referred to as a Look Modification Transform (LMT). It is mapping from one to another ACES representation. For example, a LMT can be applied to the whole sequence if a “day for night” color correction is required or a VFX pre-grade has to be done. Every shot can be still graded on its own but is viewed through the LMT and RRT/ODT.

Some camera manufacturers are supporting the IIF workflow and can setting up a LMT directly inside the camera, which is recording to scene-linear ACES data and storing the LMT in a separate file container[110]. The LMT can be changed later in the

109 Shaw, 2012, p 14
110 cf. Kainz, 2007, p 7

post production because it is not baked into the ACES pixels.

The most common way to create a LMT is using the American Society of Cinematographer's Color Decision List (ASC-CDL). Furthermore, it is possible to combine two or more LMTs, e.g., the first LMT, created within an ASC CDL, is changing the general look of the second LMT, an emulated Kodak Vision 3 stock[111].

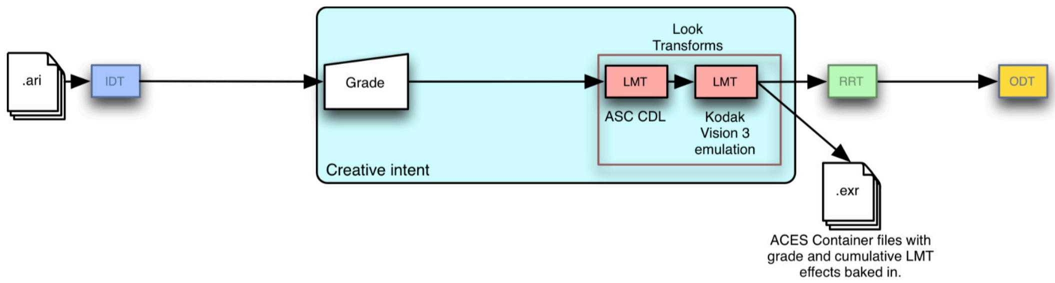


Figure 5.7: ACES Workflow with LMT

5.3 OpenEXR – The ACES File Container Format

OpenEXR is an open source image file format developed and published by Industrial Light and Magic (ILM). As earlier mentioned in chapter 3.2.2, it is the standard format when working in linear color space because of its floating point options and many additional features.

OpenEXR is the standardized image file format for the ACES image container within some missing features such as supporting only 4 channels (red, green, blue, alpha) and storing code values in only 16-bit half-precision floating point format. One big advantage of using the OpenEXR format is its 16-bit half-precision support, because it *"is compatible with the half data type in NVIDIA's Cg graphics language and is supported natively on their new GeForce FX and Quadro FX 3D graphics solutions."* [112] That means a CPU accelerated decoding of ACES files and a possible real-time playback on lower priced hardware systems.

111 cf. Technical Bulletin TB-2014-010, 2016, p 9

112 <http://www.openexr.com> (2016-08-08)

5.4 Advantages and Disadvantages of an ACES Workflow

The big advantage of the ACES workflow is its simplified and standardized color management especially in VFX pipelines. If the different facilities like DI and VFX are following the rules of the workflow, the pipeline could save time and money by preventing errors at an early stage of production. The workflow is also future proofed because of the unlimited color range of an ACES file and the different types of transformation rules, which can be modified or removed at every state for future requirements.

But there are also disadvantages such as implementation limitations of the ACES workflow in applications like After Effects or Nuke. Additional smaller problems within the ACES configuration are, for example: ACEScc is producing little noise in the blacks or the file format limits when using OpenEXR[113].

113 cf. Wheatley, 2016, p 22-24

6 Practical Example – setting up OCIO with ACES

The chapter is showing a practical example of an OCIO configuration and how it could be used in a post production pipeline with emphasis on visual effects. Besides, the given example is using the ACES workflow to demonstrate its benefits at the image interchange process during post production. The given example is using the ACEScg and ACEScc color spaces as working spaces during the whole VFX process. It is a future proofed method when working with a consistent color management in every project and already used, in a more complex form, by companies like Framestore[114].

At the first state, a possible DI workflow in DaVinci Resolve[115] is demonstrated by converting the original footage to ACEScc color space and creating a VFX pre-grade. In the next step, the OCIO/ACES configuration is created and shown by taking a look to the folder structure and the config.ocio file. The chapter continues with an overview of the OCIO toolset before showing a more detailed compositing workflow in Nuke. The last part is presenting an example of a possible matte painting workflow in Photoshop. For that reason, a preview look is baked to an ICC profile by using the OCIO tools.

6.1 The Source Footage

For the practical example, I used a shot from ***Painting Doors***[116], a short documentary, within visual effect parts, that was shot by students of the “Hochschule der Medien” in October 2015 in Germany. . The shot is showing a dream scenario in which a mother and her daughter are floating with a rubber boat through a road. The background plate was shot with the ***Sony FS7*** camera and recorded with a Slog3 image encoding curve. The foreground plate was shot in a studio in front of a green screen with the ***ARRI Alexa*** camera at EI400 and its own LogC image encoding curve[117]. I decided to use this shot because it combines different tasks in a visual effects process such as using pre-grades, merging two plates or creating matte paintings. It is also helpful for demonstrating the power of a OCIO/ACES workflow because of the different cameras and image interchanges.

114 cf. Wheatley, 2016, p 20

115 cf. <https://www.blackmagicdesign.com/de/products/davinciresolve> (2016-08-12)

116 cf. <http://www.imdb.com/title/tt5486448/?ref_=nm_knf_t4> (2016-07-20)

117 Note: We used an Exposure Index (EI) of 400 to achieve lower noise artefacts which provide better keying results in post production.

6.2 The DI Workflow

As described earlier in this paper, the DI company is responsible for the creation of the VFX pre-grades. My intention for this example was to focus on the VFX process. Thus, the following DI example has the purpose of demonstrating the creation and exchange process of a VFX pre-grade in a simplified way.

In the first step, the original footage was imported into DaVinci Resolve[118], which is a grading tool developed by Blackmagic Design[119]. Inside the application, the color management was set to ACEScc what means that the software internally converts all media files into ACEScc as a working space by choosing the right IDT. Thus, the grade is done in the ACEScc color space but displayed with the selected RRT/ODT. I specified Sony's Slog3 IDT for the background plate and Rec.709 as output transform (see *Figure: 6.1*). The grade was done with a few simple color corrections and exported as a 3D cube LUT for the later use as a preview LUT inside Nuke and Photoshop.

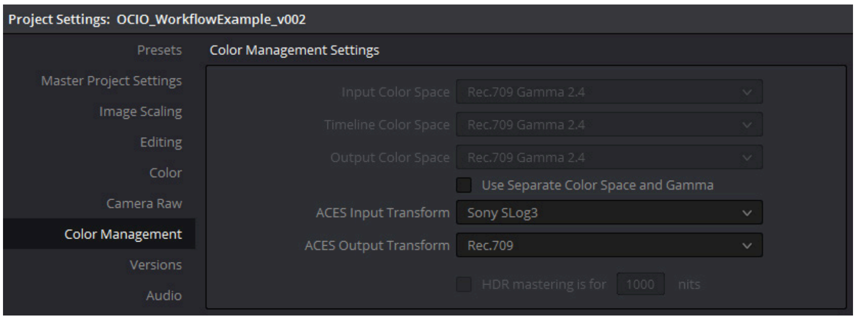


Figure 6.1: Choosing the right IDT and RRT/ODT in DaVinci Resolve

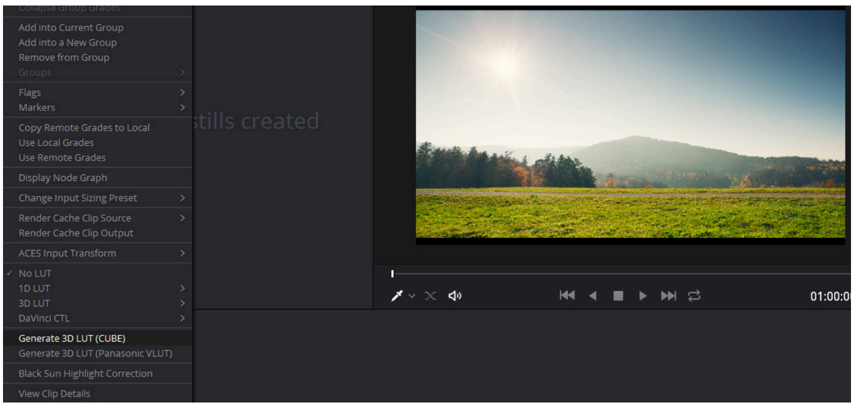


Figure 6.2: Exporting the grade as a 3D LUT in DaVinci Resolve

118 version 12.5

119 Note: Fortunately, Resolve has only a few limitations within the free version and fully supports the ACES 1.0.1 standard.

6.3 OCIO Configuration with ACES 1.0.1

Since its release of version 1.0, ACES is strongly integrated in the most common post production software tools as well as in the official OCIO-Configs repository. For the practical example, ACES 1.0.1 was employed because it supports the ACEScc color space which was used in the DI process as well as the ACEScg color space, applied as the linear working color space in Nuke.

6.3.1 Folder Structure

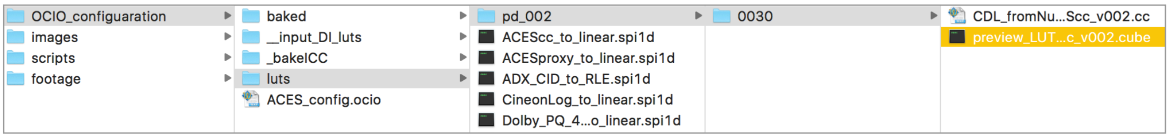


Figure 6.3: Folder structure of my example config

The ***aces_1.0.1*** sample config provided by the Academy of Motion Picture Arts and Sciences was used as a starting point for my OCIO configuration. The sample config contains all necessary LUTs in the "luts" folder and already defines the needed color spaces and roles for the ACES workflow in the config.ocio file. ***Figure 6.3*** is showing a cleaned version of the sample config folder structure in which some unnecessary LUTs were deleted. Additional folders within the VFX pre-grade LUTs for the example shot were added and specified in the ***ACES_config.ocio*** by the *search_path* command[120]:

```
search_path: luts/pd_002/0030:luts
```

6.3.2 The OCIO Configuration File

The OCIO setup is controlled by a YAML text document which has the ending ***".ocio"***. It is located on top-level at the OCIO config. In the practical example, the file is named ***"ACES_config.ocio"*** and is basically a cut-down version of the aces_1.0.1 sample config with some additional "looks" definitions applied[121].

120 Note: In bigger productions, each project has its own OCIO configuration and the folder structure can be specified with variables for sequences and shots to locate to the right direction when opening different files, e.g.: *"search_path: luts:luts/${SEQ}/${SHOT}:luts/_main"*.

121 Note: I deleted all unnecessary color space definitions, roles and aliases, e.g., all redundant IDT transform specifications.

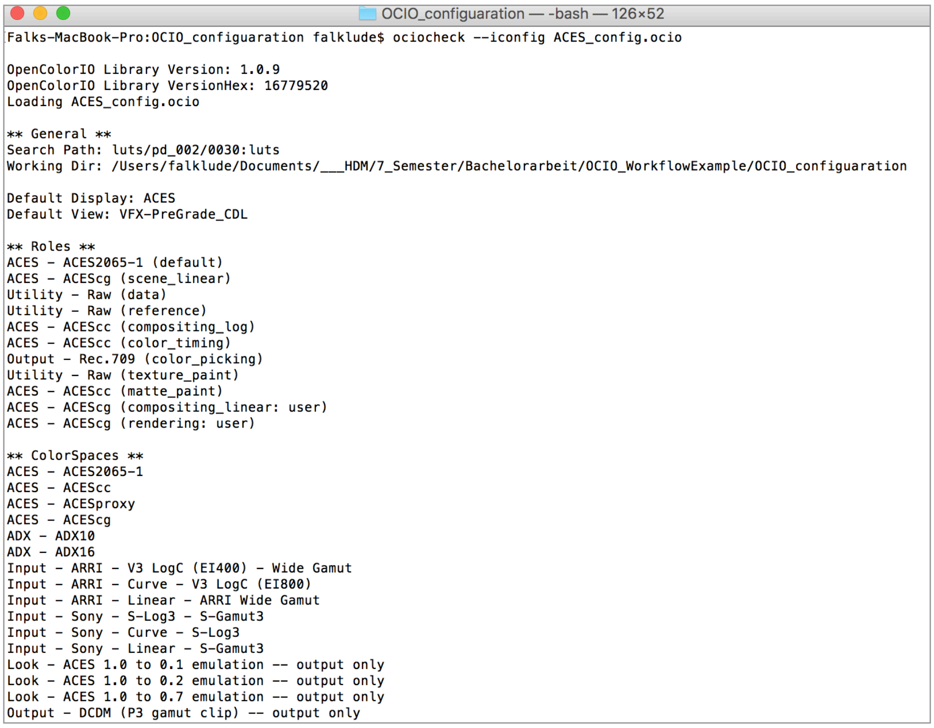
```
ocio_profile_version: 1

search_path: luts/pd_002/0030:luts
strictparsing: true
luma: [0.2126, 0.7152, 0.0722]

description: An ACES config for OCIO_WorkflowExample by Falk Lude

roles:
  color_picking: Output - Rec.709
  color_timing: ACES - ACEScc
  compositing_linear: ACES - ACEScg
  compositing_log: ACES - ACEScc
  data: Utility - Raw
  default: ACES - ACES2065-1
  matte_paint: ACES - ACEScc
  reference: Utility - Raw
  rendering: ACES - ACEScg
  scene_linear: ACES - ACEScg
  texture_paint: Utility - Raw

displays:
  ACES:
    - !<View> {name: VFX-PreGrade_CDL, colorspace: Output - Rec.709,
              looks: dailies_CDL}
    - !<View> {name: VFX-PreGrade_LUT, colorspace: ACES - ACEScc,
              looks: dailies_LUT}
    - !<View> {name: sRGB, colorspace: Output - sRGB (D60 sim.)}
    - !<View> {name: P3-DCI, colorspace: Output - P3-DCI}
    - !<View> {name: Rec.709, colorspace: Output - Rec.709}
    - !<View> {name: Raw, colorspace: Utility - Raw}
    - !<View> {name: Log, colorspace: ACES - ACEScc}

active_displays: [ACES]
active_views: [VFX-PreGrade_CDL, VFX-PreGrade_LUT, sRGB, P3-DCI,
              Rec.709, Raw, Log]

looks:
  - !<Look>
    name: dailies_CDL
    process_space: ACES - ACEScc
    transform: !<FileTransform> {src: CDL_fromNuke_ACEScc_v002.cc,
               interpolation: linear}

  - !<Look>
    name: dailies_LUT
    process_space: ACES - ACEScc
    transform: !<FileTransform> {src: preview_LUT__ACEScc_v002.cube,
               interpolation: linear}
```

Figure 6.4: Part of the ***ACES_config.ocio*** *example file*

6.4 The OCIO Toolset

OpenColorIO comes with a few handy tools. This part of the chapter is giving a basic overview of their functionalities and how to use them.

OCIOCHECK This command line tool is showing an overview of a specified OCIO config file. It can become very useful when working with a complex OCIO configuration.



```
OCIO_configuaration — -bash — 126×52
Falks-MacBook-Pro:OCIO_configuaration falklude$ ociocheck --iconfig ACES_config.ocio

OpenColorIO Library Version: 1.0.9
OpenColorIO Library VersionHex: 16779520
Loading ACES_config.ocio

** General **
Search Path: luts/pd_002/0030:luts
Working Dir: /Users/falklude/Documents/___HDM/7_Semester/Bachelorarbeit/OCIO_WorkflowExample/OCIO_configuaration

Default Display: ACES
Default View: VFX-PreGrade_CDL

** Roles **
ACES - ACES2065-1 (default)
ACES - ACEScg (scene_linear)
Utility - Raw (data)
Utility - Raw (reference)
ACES - ACEScc (compositing_log)
ACES - ACEScc (color_timing)
Output - Rec.709 (color_picking)
Utility - Raw (texture_paint)
ACES - ACEScc (matte_paint)
ACES - ACEScg (compositing_linear: user)
ACES - ACEScg (rendering: user)

** ColorSpaces **
ACES - ACES2065-1
ACES - ACEScc
ACES - ACESproxy
ACES - ACEScg
ADX - ADX10
ADX - ADX16
Input - ARRI - V3 LogC (EI400) - Wide Gamut
Input - ARRI - Curve - V3 LogC (EI800)
Input - ARRI - Linear - ARRI Wide Gamut
Input - Sony - S-Log3 - S-Gamut3
Input - Sony - Curve - S-Log3
Input - Sony - Linear - S-Gamut3
Look - ACES 1.0 to 0.1 emulation -- output only
Look - ACES 1.0 to 0.2 emulation -- output only
Look - ACES 1.0 to 0.7 emulation -- output only
Output - DCDM (P3 gamut clip) -- output only
```

Figure 6.5: OCIOCHECK in OSX terminal

OCIOBAKELUT This command line tool is giving the opportunity to bake out a LUT in a supported LUT format[122] by specifying an OCIO config file and setting an input- and output-color space.

OCIOCONVERT The tool is loading an image file, applying a color transform and saving to a new file. For using the tool, OpenImageIO must be installed[123]. It can become very useful to automate conversions for e.g., textures, which are used in CG or matte paintings.

122 Supported LUT formats: cf. <http://opencolorio.org/FAQ.html#faq-supportedlut> (2016-08-12)

123 cf. <https://sites.google.com/site/openimageio/home> (2016-08-12)

OCIODISPLAY This is a basic image viewer which uses OpenImageIO to load an image and displays it by using OCIO. The viewer is providing some additional controls (exposure and gamma)[124].

6.5 The Compositing Workflow

While compositing, all images created by the different departments (CGI, FX, matte paintings) come together and result to the final image. Thus, it is the most important part in the VFX pipeline in terms of dealing with colors because everything has to fit together.

6.5.1 OCIO with The Foundry's Nuke

The most common compositing tool nowadays is The Foundry's Nuke, which is also used in this practical example. For the practical example I was working with Nuke 9, which already provides a lot of useful OCIO tools explained later in this chapter. Nevertheless, a few things were still missing like the support of an OCIO config in the Read or Write nodes[125]. Another OCIO implementation issue, which can produce noise artefacts at some operations, appears when working in the ACEScc color space in Nuke[126].

6.5.1.1 Loading the OCIO Config File

The first step of the Nuke-OCIO workflow was loading my ACES_config.ocio file into Nuke. This could be done manually by just selecting the file in Nuke's preferences. In my example I was using the environment variable "OCIO" and setting its value to the *ACES_config.ocio* location path. Thus, the OCIO configuration is selected automatically in every new Nuke script.

124 cf. <http://opencolorio.org/userguide/tool_overview.html#ociodisplay> (2016-08-12)

125 Note: "With the introduction of Nuke 10.0, the Foundry have extended the capabilities of the Read and Write nodes to allow the colour space conversions to be defined using OpenColorIO." [Wheatley, 2016, p 22]

126 cf. Wheatley, 2016, p 22

6.5.1.2 OCIO Tools in Nuke

As mentioned earlier in this chapter, Nuke already provides a few powerful OCIO nodes. These nodes can be used after loading the OCIO config file:[127]

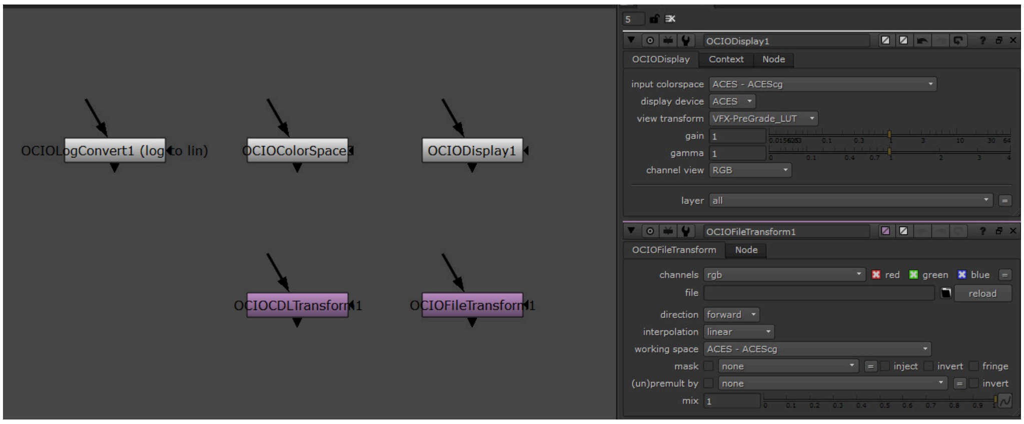


Figure 6.6: OCIO nodes in Nuke

OCIOColorSpace The node is used for OCIO color space conversions like the Nuke's standard Colorspace node.

OCIOLogConvert The node is converting from compositing_log to scene_linear and vice versa. Since those color spaces are defined by the OCIO config file in the "roles" section, an artist can't do something wrong and uses always the right color spaces.

OCIODisplay The node is an implementation of the external OCIO image viewer with the same exposure and gamma settings and the opportunity to display the view transforms specified in the OCIO config file.

OCIOFileTransform The node is used for applying an external LUT file (e.g. a 1D or 3D LUT) to a specific input.

OCIOCDLTransform This node is giving the opportunity to either load or create a CDL transform and export it for the use in other applications or exchange with other companies.

127 Note: The nodes could also be used without a loaded OCIO config, but in this case it wouldn't make a difference to the standard Nuke nodes.

6.5.1.3 Converting the Plates

As described in chapter 3.2.2, the footage has to be converted into a linear light space for VFX purposes. In the given example, the original plates were converted first into ACES2065-1 color space by applying Sony's and ARRI's IDTs and stored into the OpenEXR format (see ***Figure 6.7***). The ACES2065-1 encoding was used to ensure a good exchange between the post production companies. To ensure a manageable workflow especially when rendering CGI, the ACES2065-1 images (AP0 primaries) were transformed into ACEScg with its AP1 primaries.

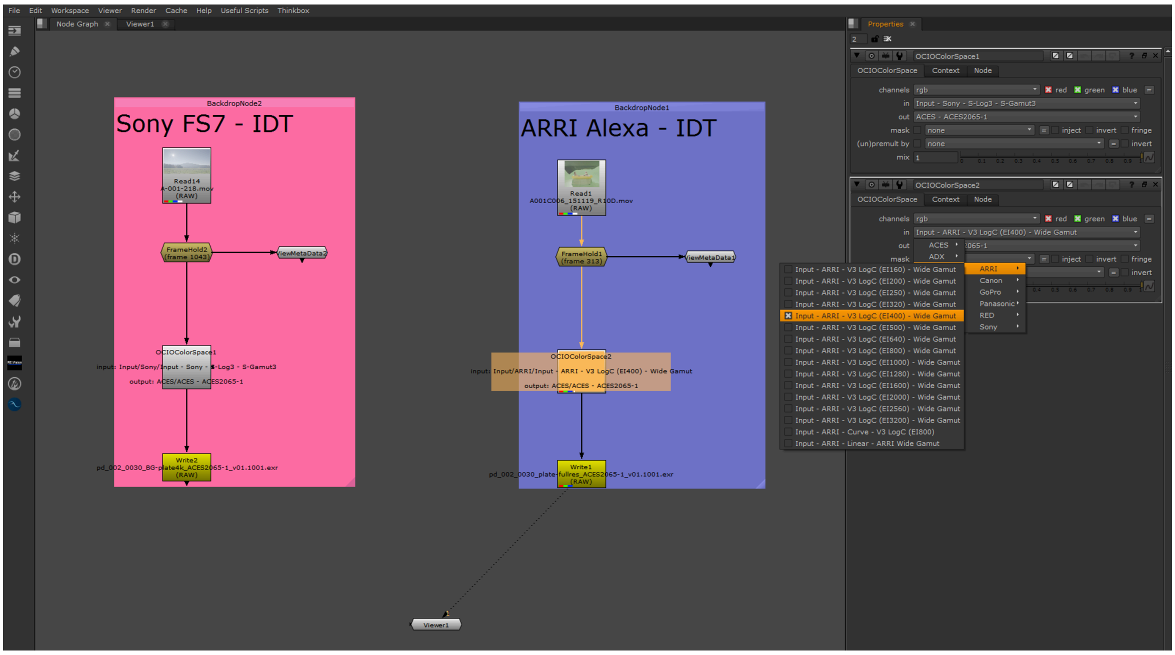


Figure 6.7: Converting the Plates into ACES2065-1

6.5.1.4 Working with Nuke's Viewer Process

When working in Nuke with a linear color encoded ACEScg image, a viewer process is necessary for evaluating correctly the image. This part of the chapter is describing the workflow of setting up Nuke's viewer process within the preview LUT created by the DI facility.

At the first step, the incoming LUT was copied into the folder structure of the OCIO configuration. To apply a new viewer process in Nuke when using OCIO, a "view" in the display section has to be specified. It requires a name and a color space statement and can be supplemented with a "look" that is defined later in the config. For example, the command *- !<View> {name: Rec.709, colorspace: Output - Rec.709}* makes it possible to display the linear encoded image in Rec.709 color space by selecting the “Output – Rec.709” RRT/ODT color space defined in the OCIO config file as the active viewer input process in Nuke.

As previously described, in the practical example the grade was applied in ACEScc color space and processed with the Rec.709 RRT/ODT before baked out as a 3D LUT. Thus, the look and display view in the *ACES_config.ocio* file were defined as followed:

```
displays:
  ACES:
- !<View> {name: VFX-PreGrade_LUT, colorspace: ACES - ACEScc,
          looks: dailies_LUT}

looks:
  - !<Look>
    name: dailies_LUT
    process_space: ACES - ACEScc
    transform: !<FileTransform> {src: preview_LUT__ACEScc_v002.cube,
               interpolation: linear}
```

Figure 6.8 is showing the viewer input process *VFX-PreGrade_LUT* (Viewer2), defined in the *ACES_config.ocio* file, compared with a manual setup (Viewer1) using the *OCIOFileTransform* node to load the 3D LUT. Another common way to create VFX pre-grades in the DI/VFX pipeline is working with an ASC Color Decision List (CDL), which I also tested and defined in the final *ACES_config.ocio* file (see *Figure 6.4*)

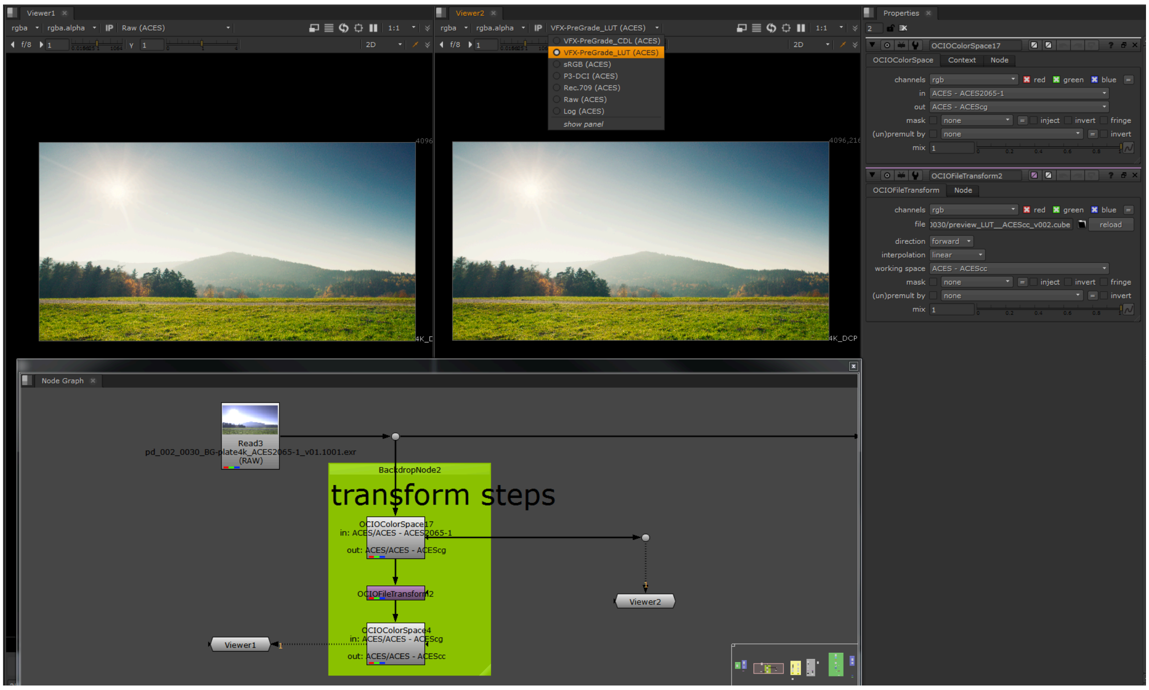


Figure 6.8: Converting the plates into ACES2065-1

6.6 The Matte Painting Workflow

In the last part of the practical example, a possible image interchange workflow is shown between compositing and matte painting. Matte Painters are generally working with sRGB or Log files in Photoshop. In this example, the ACEScc color space was used for the matte painting to demonstrate an easy interchange within the ACES workflow.

6.6.1 Baking Look to ICC Profile

To ensure that both departments are working with the same color appearances, the VFX pre-grade was baked into an ICC profile. Therefore, Photoshop was using the same film emulation transform as Nuke. The ICC profile was generated with the *OCIOBAKELUT* command executed in the Mac OS X Terminal application:

```
ociobakelut
--format icc
--inputspace 'ACES - ACEScc'
--looks dailies_LUT
--outputspace 'ACES - ACEScc'
--displayicc _bakeICC/monitorprofiles/140819_DISPLAY_EV2436_v03_D65.icm
--description "ACEScc to Preview_Lut"
baked/photoshop/ACEScc_to_Preview_Lut_EIZO_v01.icc
```

As seen, the indication of an input and output color space specified in the OCIO config file was required and the additional VFX pre-grade look was applied between the input and output transforms. Furthermore, the tool needed a reference ICC profile specified in the –displayicc command to match the new ICC profile to the reference. In this example, I used the currently active ICC profile of my EIZO monitor[128] which was created in a calibration software. If no specific monitor profile is used on the active display, the –displayicc command can be omitted. In this case the standard "sRGB IEC61966-2.1" profile is used by default[129].

In this example, the *OCIOBAKELUT* tool was used in a simple way. However, with a bit more coding, it should be possible to automate this process and make it "production ready" for big projects with many shots.

128 140819_DISPLAY_EV2436_v03_D65.icm

129 cf. <http://opencolorio.org/userguide/baking_luts.html> (2016-08-13)

6.6.2 Nuke to Photoshop Workflow

In order to receive a good starting point for a matte painting, a reference frame is usually created by the compositor or matte painter. Thus, the image was converted in Nuke from ACEScg to ACEScc and saved as a 16-bit TIFF file to guarantee the use of all available tools in Photoshop as described in chapter 4.2.3.

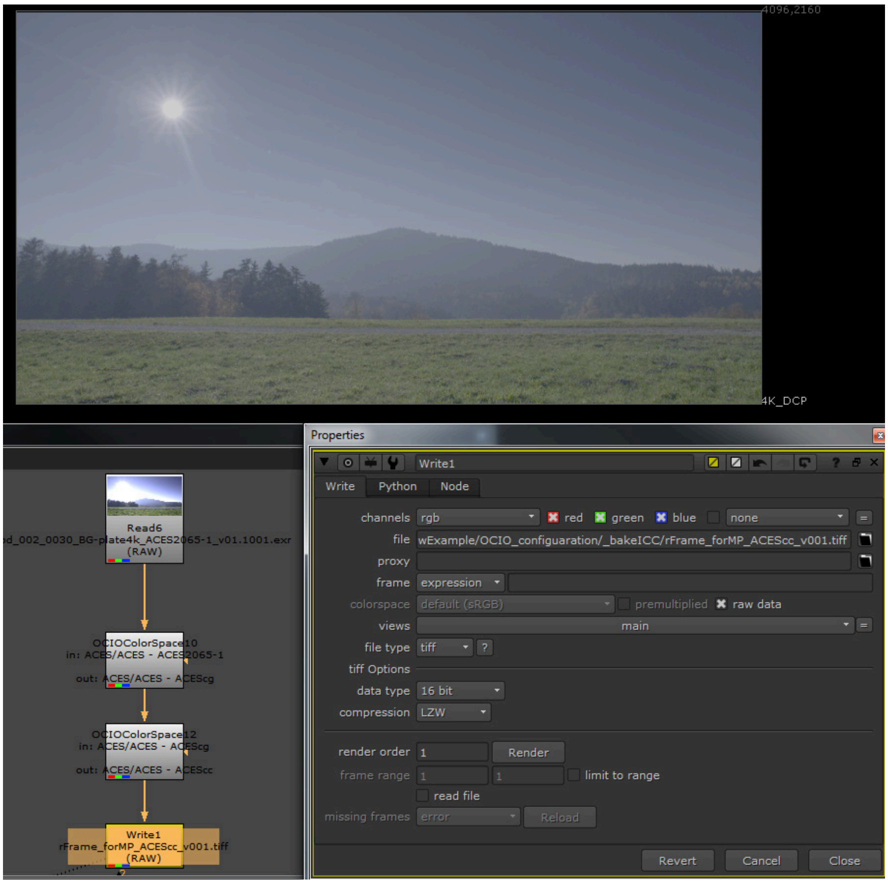


Figure 6.9: Converting and saving the image to ACEScc logarithmic color space in Nuke

After loading the reference frame in Photoshop, a custom proof condition was applied with the generated VFX pre-grade ICC profile. Within this setup I was able to work in the ACEScc color space but looking through the VFX pre-grade. *Figure 6.10* is comparing the displayed images in Nuke and Photoshop and showing the proof-settings in Photoshop[130].

If the matte painting process is finished, the file could be saved either as Photoshop or TIFF file and is hand-offed to the compositing department. In the given example, a

130 Note: Depending on which color space is used for the matte painting, other proof-settings should be applied to achieve the best match.

few big mountains were added to the background by converting a reference texture from sRGB color space to ACEScc logarithmic color space and importing it into the Photoshop matte painting file. This conversion could also be realized and automated by using the OCIOCONVERT tool.

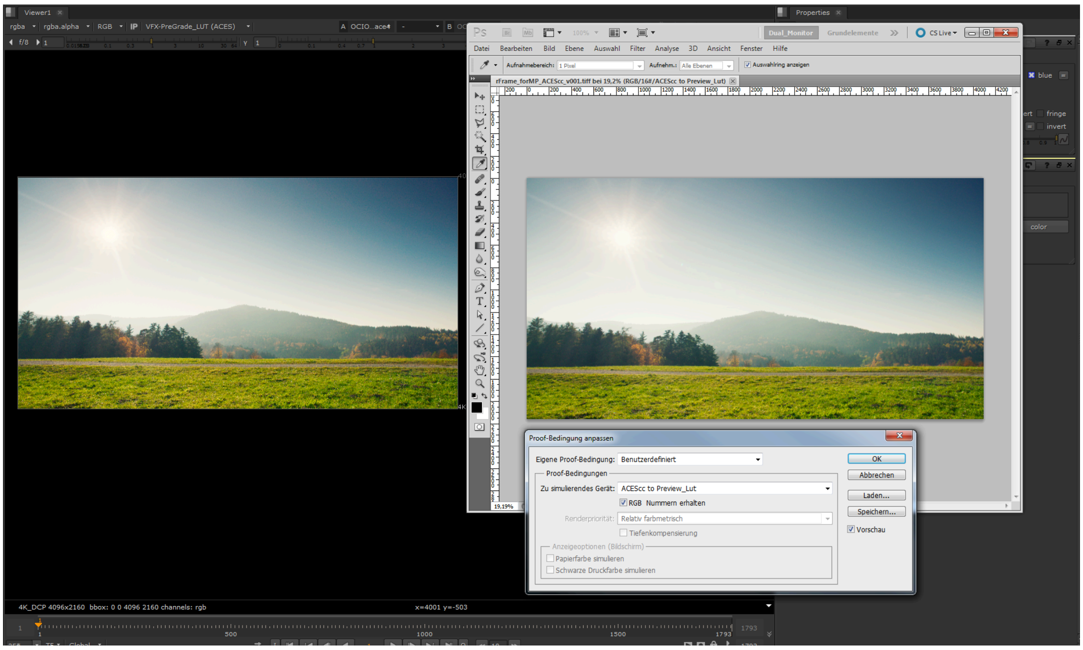


Figure 6.10: Color proof setup in Photoshop

In the last step, the matte painting was loaded into Nuke, and a color space conversion was applied to continue working in ACEScg linear light space (*Figure 6.11*).

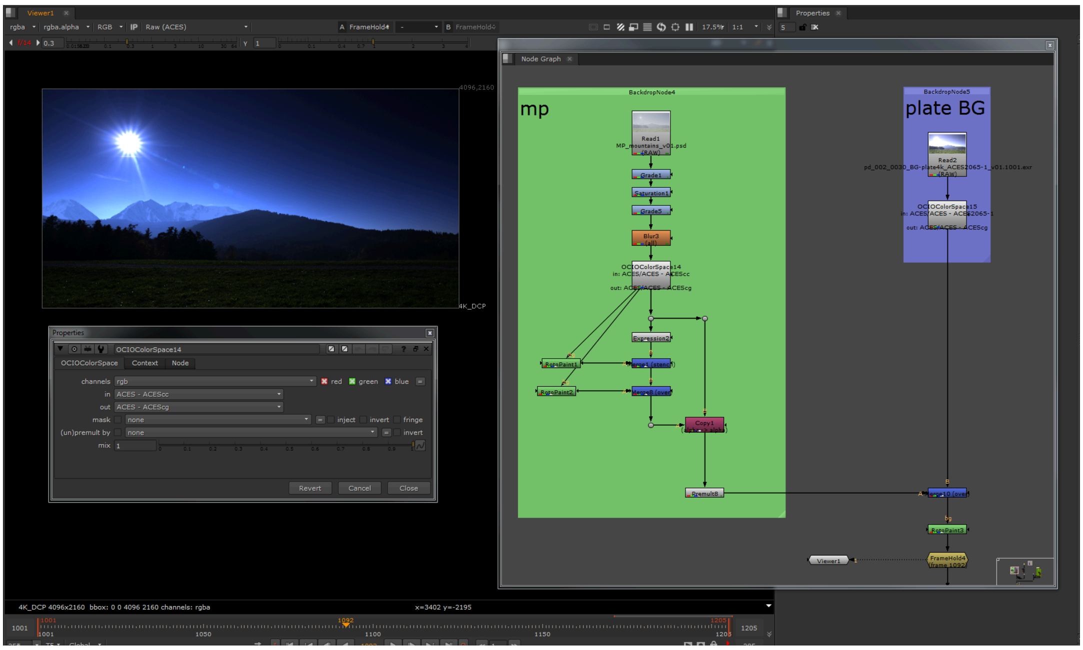


Figure 6.11: Matte painting integration in Nuke

Figure 6.12: Final comp - transformed into Rec.709 color space

Figure 6.13: Final image - the rendered final comp (EXR) is graded in ACEScc color space and transformed with Rec.709 RRT/ODT

7 Conclusion and Outlook

Within the implementation of the ACES 1.0.1 standard, OpenColorIO gets more important in the field of the whole post production process besides the normal VFX area, because the image interchange between DI and VFX facilities became more standardized when using the IIF/ACES workflow. Since the Academy released ACES 1.0 within the useful color encodings ACEScg and ACEScc, the workflow has become more interesting for VFX companies when using those encodings as working color spaces in compositing as well as CG and matte painting. The power of these two new color spaces in comparison to the standard ACES2065-1 color space (AP0 primaries) is the more flexible encoding with the AP1 primaries, which means no CG rendering issues and a strong exchange between ACEScg and ACEScc without losing color information.

OpenColorIO combined with ACES is a future proofed workflow which already works really well together. However, the configuration of the ACES color spaces in addition with other color transforms can become very complex and time consuming. Furthermore, the issues of ACEScc and the file format limits described in chapter 5.4 are still common problems. Besides, for future releases of the OCIO Core Library, it would be nice to have the opportunity of simple if-else statements. This could be useful to organize the color spaces better by using two or more different LUTs within the same folder structure (e.g., when shooting with different cameras in one shot).

The success of ACES as a framework will depend either on how many DOPs are going to shoot movie pictures in ACES color space directly, or how many DI companies and colorists will use the ACES framework in their grading pipeline. Unfortunately, many colorists are using the powerful AlexaV3LogC encoding as their working color space and thus, the ACES specification is still not as common as it should be from the perspective of the Academy and many VFX companies.

In summary, OpenColorIO is a very powerful color management tool that is in use at many VFX companies today. With its flexible configuration structure and the possibility of using it in many VFX-relevant software application, OCIO meets the most requirements of a modern post production pipeline. Furthermore, it is easy to enhance the tools or the implementation in a specific pipeline, because of its C++ and Python availability and works in a smooth and stable way with the ACES standard.

Bibliography

ARRI ALEXA Color Processing phite paper, Arnold & Richter Cine Technik, 2011. <https://www.arri.com/?eID=registration&file_uid=7347>, (2016-07-27)

Duiker, Haarm-Pieter; **ACES 1.0 OpenColorIO config - Siggraph 2015,** 2015. <http://www.slideshare.net/hpduiker/aces-10-opencolorio-config-siggraph-2015>, (2016-08-15)

Häßler, Ulrike; **Vom RGB-Dreieck zum LAB-Farbraum,** 2008. <http://www.wisotop.de/CIE-Normfarbtafel.shtml>, (2016-07-25)

Hunt, Robert; Pointer, Michael; **Measuring Colour**, 4th ed., Wiley, 2011.

Kainz, Florian; **Using OpenEXR and the Color Transformation Language in Digital Motion Picture Production,** Industrial Light & Magic, 2007. <http://www.openexr.com/UsingOpenEXRandCTL.pdf>, (2016-07-25)

Meyer, Chris; Meyer, Trish; **Linear Light**, CyberMotion, <http://s3.artbeats.com/articles/pdf/linear_light.pdf>, (2016-08-01)

McHugh, Sean; **COLOR MANAGEMENT: COLOR SPACES,** 2016. <http://www.cambridgeincolour.com/tutorials/color-spaces.htm>, (2016-07-25)

OCIO Online Documentation, <http://opencolorio.org/userguide/config_syntax.html>, (2016-08-02)

Poynton, Charles; **Digital Video and HDTV Algorithms and Interfaces**, Elsevier Science, 2003.

Pratt, William; **Digital Image Processing: PIKS Inside,** 3rd ed., John Wiley & Sons Ltd, 2001

Poynton, Charles; **Frequently Asked Questions about Gamma,** 1998. <http://www.poynton.com/PDFs/GammaFAQ.pdf>, (2016-07-28)

Selan, Jeremy; **Cinematic Color, VES Technology Committee,** 2012. <http://github.com/jeremyselan/cinematiccolor/raw/master/ves/Cinematic_Color_VES.pdf>, (2016-07-22)

Shaw, Kevin; **The Beauty of ACES with Nucoda Film Master,** Digital Vision, 2012. <http://www.digitalvision.tv/w/images/2/26/ACES.pdf>, (2016-08-08)

The ACES Project Committee, ACES Documentation Guide, 2016. <http://j.mp/TB-2014-001>, (2016-08-04)

The ACES Project Committee, **Procedure P-2013-001,** 2016. <http://j.mp/P-2013-001>, (2016-08-04)

The ACES Project Committee, **Specification S-2014-004,** 2016. <http://j.mp/S-2014-004>, (2016-08-04)

The ACES Project Committee, **Specification S-2014-003,** 2016. <http://j.mp/S-2014-003>, (2016-08-04)

The ACES Project Committee, **Technical Bulletin TB-2014-010,** 2016. <http://j.mp/TB-2014-010>, (2016-08-04)

The Quantel Guide to Digital Intermediate, <http://www.quantel.info/resource.nsf/files/brochure_DI_Guide/$FILE/brochure_DI_Guide.pdf>, (2016-07-29)

Tooms, Michael; **Colour Reproduction in Electronic Imaging Systems: Photography, Television, Cinematography,** John Wiley & Sons Ltd, 2015.

Wheatley, Kevin; **A retrospective on the adoption of the ACES technology at Framestore,** ACM New York, 2016. <http://dl.acm.org/ft_gateway.cfm?id=2947695&ftid=1759010&dwn=1&CFID=819421336&CFTOKEN=38553917>, (2016-07-29)

Wright, Steve; **Digital Compositing for Film and Video,** 3rd ed., Focal Press, 2010.

Picture and Diagram Sources

Figure 2.1: **The Color Spectrum,** (Selan, 2012, p 7)

Figure 2.2: **The CIE 1931 color-matching functions,** (Poynton, 2003, p 216)

Equation 2.3: **Normalizing the XYZ tristimulus values**

Equation 2.4: **Recovering X and Z tristimulus values**

Figure 2.5: **The CIE 1931 [x, y] Chromaticity Diagram,** (Selan, 2012, p 11)

Figure 2.6: **The CIE u'v' Diagram,**
Source: <https://upload.wikimedia.org/wikipedia/commons/8/83/CIE_1976_UCS.png>,
© 2016 Adoniscik.

Figure 2.7: **Rec.709 and DCI-P3 color gamuts**

Figure 3.1: **Post production workflow diagram**

Figure 3.2: **The „video-centric" DI workflow**

Figure 3.3: **The „film-centric" DI workflow**

Figure 3.4: **Log-Grading example,**
Source: <http://opencolorio.org/>, (2016-08-01)
Images from *"Cloudy With A Chance of Meatballs"*
© 2011 Sony Pictures Inc. All Rights Reserved.

Figure 3.5a: **Tone characteristic of Linear encoded image,**
(ARRI ALEXA Color Processing white paper, 2011, p 12)

Figure 3.5b: **Tone characteristic of LogC encoded image,**
(ARRI ALEXA Color Processing phite paper, 2011, p 10)

Figure 4.1: **Color Space selection in Nuke's Read node**

Figure 4.2: **V-Ray OCIO support in Maya,**
Source: <https://docs.chaosgroup.com/download/attachments/12085741/image2014-1-26%20
17%3A4%3A33.png>, (2016-08-02), © 2016 Chaos Software Ltd.

Figure 4.3: **Limited Photoshop Tools**

Figure 4.4: **Install OCIO via Homebrew**

Figure 4.5: **OCIO Start Configurations**

Figure 4.6: **OCIOLogConvert**

Figure 4.7: **Family Groups in Nuke**

Figure 5.1: **Top-level ACES workflow,** (Selan, 2012, p 42)

Figure 5.2: **ACES Primaries**

Equation 5.3: **Transformation Matrix ACES to ACEScg,** (Specification S-2014-004, 2016, p 8)

Figure 5.4: **Framestore's Post Production Pipeline Diagram,** (Wheatley, 2016, p 26)

Figure 5.5a: **Print Film emulation,** (Shaw, 2012, p 11)

Figure 5.5b: **RRT curve,** (Shaw, 2012, p 11)

Figure 5.6: **The RRT/ODT workflow**
source: <http://www.digitalpreservation.gov/partners/documents/IIF_Overview_August_2010.
pdf>, (2016-08-08),
© 2010 Academy of Motion Picture Arts & Sciences Science and Technology Council

Figure 5.7: **ACES Workflow with LMT,** (Technical Bulletin TB-2014-010, 2016, p 12)

Content of the DVD

Contents of the attached DVD:

- PDF version of this thesis
- all pictures and diagrams
- the corresponding OCIO example configuration
- the corresponding rendered images