OPPO O-Log

White Paper

Document Version History		
Version	Author(s)	Change Note
2025-5-31	Zemin Lai Thom Qu	Initial document version
2025-9-29	Thom Qu	ACES
2025-10-22	Cris Yan Zoey Chow	CHN to ENG Translation

Table of Contents

Preface

Log video is designed to maximize the retention of video information, resulting in an effect similar to the RAW image format in photography after video preprocessing (Log video undergoes simple processing and is more comparable to the DNG format in photography).

Due to the significantly larger volume of raw image data processing required for video clips compared to single photos, specialized hardware modules are typically employed to assist in processing. Current devices integrate dedicated hardware image processing modules at the shooting end, so Log video undergoes some preliminary processing upfront (Log video is not merely a raw compression of each video frame but involves basic pre-processing of the video raw data).

The video image data processing utilizes a logarithmic curve and a wide color gamut. By aligning the human perception of brightness (which follows an exponential pattern) with sensor characteristics (which follow a linear pattern), rithmic-based Log curve is used for compression (14-bit original linear data and considering the data bit depth of the sensor image [10-bit to 14-bit], a logarithmic-based Log curve is used for compression [14-bit original linear data is compressed into 10-bit Log data. This step is lossy compression.])

The purpose of using the Log video configuration is to preserve the maximum amount of video data with minimal processing during capture. Log video is not suitable for direct viewing but is intended as source footage for use in post-production workflows.

1 Introduction

This document describes the OPPO O-Log logarithmic color space (referred to as O-Log) and its application scenarios.

OPPO O-Log is specifically designed to enhance the encoding precision and production suitability of dynamic range for small sensors in mobile imaging.

The O-Log curve is a logarithmic image encoding method used in OPPO mobile devices. This encoding exhibits grayscale characteristics similar to human visual perception. However, due to fundamental differences between digital cameras and human vision, their color characteristics remain distinct.

Logarithmic encoding means that, across a wide range, there is a linear (straight-line) relationship between exposure values measured in stops and the encoded signal. Each increase of one stop in exposure results in a consistent increase in signal value. The slope of this segment of the curve is referred to as the gamma value. A toe region can be observed at the bottom of the curve, which exists because the sensor's quantization accuracy for low light levels cannot match that of highlight regions. The overall shape of this curve closely resembles the human perceptual response curve.

2 Hardware Encoding

This section outlines the O-Log hardware encoding curve utilized internally within the camera. The hardware specifications provided are intended solely as reference material related to O-Log sensor signal encoding and are not intended for software implementation.

2.1 Encoding Format

To balance storage and encoding efficiency, OPPO O-Log video employs lossy compression encoding (H.265 YUV 4:2:0).

To facilitate secondary color grading of the footage in post-production, O-Log video processing utilizes a universal wide-gamut BT.2020 transformation matrix.

2.2 Encoding Quality

Video encoding utilizes a dynamic bitrate, with a reference standard of 120Mbps (±20%) for 4K 60fps encoding.

The approximate storage space required for one minute of O-Log video recording is as follows:

1080P 30fps: ~240MB
1080P 60fps: ~360MB
1080P 120fps: ~540MB
4K 30fps: ~600MB
4K 60fps: ~900MB
4K 120fps: ~1350MB

3 OPPO O-Log Curve

This section formally defines OPPO O-Log — a scene-referred logarithmic color space composed of the OPPO O-Log transfer function and the OPPO Wide Gamut (hereinafter referred to as O-Gamut) primary system.

3.1 Encoding Function

The O-Log curve is essentially a family of curves corresponding to different exposure indices (EI). In default auto-exposure, a scene element with 18% reflectance is mapped to an O-Log value of 0.4901589 (approximately 502/1023). The figure below presents the O-Log curve in a linear-linear coordinate system, where the horizontal axis represents the sensor signal (normalized to 1, corresponding to a 10EV input signal).

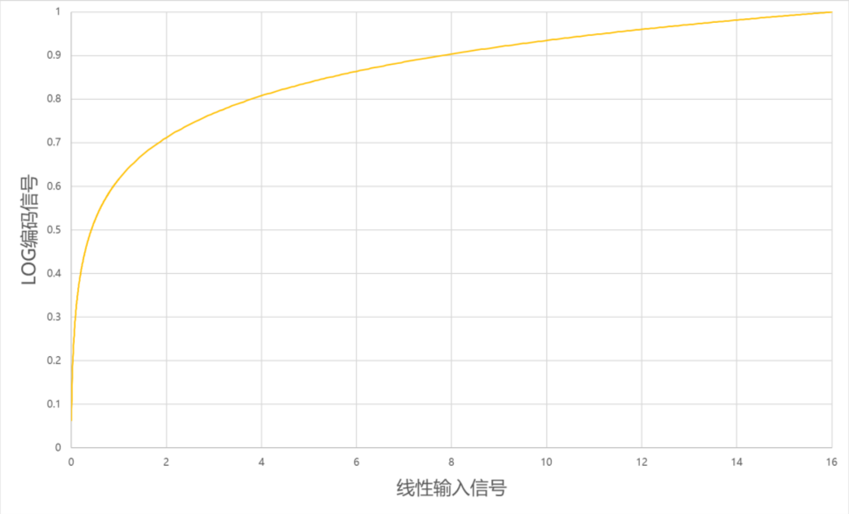

Figure 1：The hardware encoding curve applied to the linear signal.

X axis： Linear Input Signal ,Y axis：Log Encoded Signal

Scene reflection	Linear Signal	Log Signal	
		Normalized Floating-Point Values	10 bit values（full range）
0%	0	0.0631271	64
18%	0.01125	0.3895463	399
39%	0.0244	0.4901589	502
1600%	1	1	1023

Table 1: Log Encoding Curve for Linear Input Signal

The compression curve used for Log video in practical scenarios is dynamically adjusted based on scene luminance. For clarity, the figure below displays only one full-range Log curve.

The encoding function is defined as:

$$P = f(R) = \gamma \cdot \log_e(R + \beta) + \delta, \quad 0 \le R \le 16$$

where:

$$\gamma = 0.139$$
$$\beta = 0.019$$
$$\delta = 0.614$$

e is the base of the natural logarithm, approximately 2.7182818

A single-segment function is used to complete the entire encoding mapping.

3.2 Decoding Function

The decoding function, which restores Log image data to linear scene reflectance signals, is defined by the following equation:

$$R = f^{-1}(P) = \exp((P - \delta) / \gamma) - \beta,$$
$$\text{where } R_0 \le P \le 1$$

and:

$$R_0 = 0.0631271$$
$$\gamma = 0.139$$
$$\beta = 0.019$$
$$\delta = 0.614$$

e denotes the base of the natural logarithm, approximately 2.7182818.

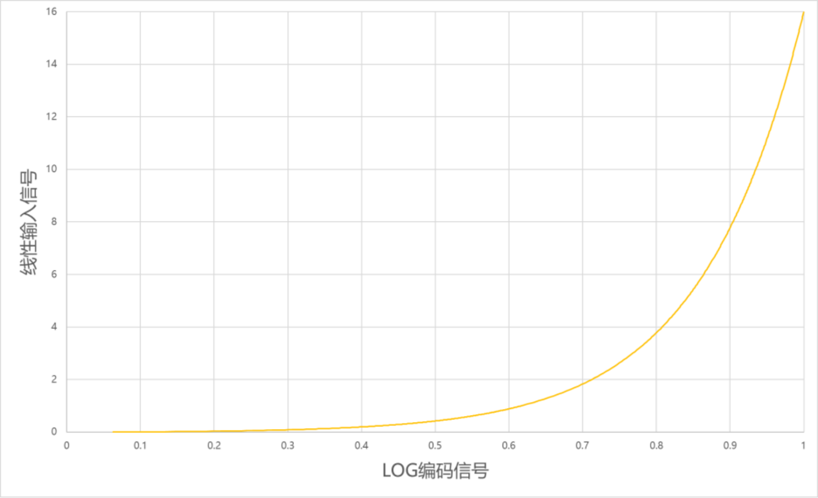

Figure 2: Hardware Decoding Curve for Log Signal

X axis：Log Encoded Signal , Y axis：Linear Input Signal

4 Color Space (OPPO Wide Gamut)

The color space describes how colors are represented. OPPO O-Log uses RGB primaries and color space conversion matrices as defined in the ITU-R BT.2020-2 specification. The white point is set to the D65 illuminant. And all chromaticity values are specified using the CIE 1931 2° standard observer colorimetry.

CIE chromaticity coordinates		
	X	Y
Red primary (R)	0.708	0.292
Green primary (G)	0.170	0.797
Blue primary (B)	0.131	0.046
White point (D65)	0.3127	0.3290

The conversion between RGB and luminance-chrominance (YCbCr) representations is defined by the following equations:

$$Y = 0.2627 * R + 0.6780 * G + 0.0593 * B$$

$$C_b = (B - Y) / 1.8814$$

$$C_r = (R - Y) / 1.4746$$

5 Dynamic Range

5.1 Scene-Referred Dynamic Range

Tonal values in an image, in physical terms, represent surfaces in a scene that reflect or emit varying amounts of light. Surface brightness is measured as luminance, for which the SI unit is the candela per square meter (cd/m²). In China, the synonymous term "nit" is commonly used. When the brightest object has a luminance of 4096 cd/m² and the darkest object has a luminance of 0.5 cd/m², the scene's contrast or dynamic range is 8192:1 (4096:0.5). Photographers and cinematographers measure this ratio in stops; in this example, it is 13 stops ($2^{13} = 8192$).

A spot meter measures the physical quantity of luminance, but it presents the result as a combination of aperture and exposure time corresponding to a selected ISO sensitivity. The absolute brightness of the scene is relatively less critical, as it can be compensated for in several ways: by adjusting the aperture, using neutral density (ND) filters, or changing the camera's exposure index (EI) setting.

If the luminance of both objects in the example above were doubled (8192:1 instead of 4096:0.5), the contrast ratio would remain unchanged. Closing the aperture by one stop would have the same effect on the tonal range. The relative luminance of elements in the scene translates linearly to exposure on the sensor—the photosites receiving the most light will collect 8192 times more photons than those receiving the least light.

5.2 Sensor Dynamic Range

The dynamic range of a sensor is defined as the ratio between its saturation signal and sensitivity threshold. This concept is not pioneered by OPPO but aligns with general engineering practices outlined in standards such as ISO 15739 or the EMVA Standard 1288.

Engineers typically measure a sensor's dynamic range based on linear raw signal values. However, cinematographers are more concerned with the camera's dynamic range, making it more practical to consider the entire camera image processing pipeline. Most cameras process images into a logarithmic-like RGB encoding (e.g., OPPO uses the O-Log curve). This leads to another challenge when deriving dynamic range through visual inspection of test charts: different manufacturers employ different encoding curves (which may even vary across versions), making direct comparisons of measurements challenging.

To perform quantitative measurements of the signal-to-noise ratio (SNR), image data must be converted back to the linear domain. The officially published dynamic range values for OPPO cameras are obtained using this method: images are first processed into O-Log format, and then linearized using the inverse of the corresponding curve. Based on this linearized data, the SNR for each exposure is calculated to determine the sensitivity threshold and saturation point. It should be noted that due to technical constraints of small-sized sensors, dynamic range may vary across different sensors, resolutions, and frame rates. The table below presents data based on the maximum dynamic range.

Products	Dynamic Range（SNR=1）
OPPO Find X8 Ultra	14 stops
OnePlus 15	13 stops
OPPO Find X9	14 stops
OPPO Find X9 Pro	14 stops

Engineers define the sensitivity threshold as the lower signal limit at which the noise level equals the signal itself. For cinematographers, however, such a high relative noise level may be unacceptable. Beyond the dynamic range of the sensor, other characteristics must be considered — for example, shadows should exhibit no linear noise patterns or color cast.

Some users prefer to use SNR = 2 as the lower threshold for dynamic range measurement. This criterion results in a value that is at least 1 stop lower than the nominal value obtained with SNR = 1.

6 ACES transform

The ACES conversion is performed using a 3×3 matrix as required by the ACES standard. The corresponding CTL code is provided below:

```
// O-Log REC2020 to ACES AP0 CTL

//define rec2020_to_xyz make_mat3(make_float3(0.6370f, 0.1446f, 0.1689f), make_float3(0.2627f, 0.6780f, 0.0593f),

make_float3(0.0f, 0.0281f, 1.0610f))

//define d65_to_d60_cat02 make_mat3(make_float3(1.01174414f, 0.00770577991f, -0.0157216747f),

make_float3(0.00555788933f, 1.00153586f,-0.00626219941f), make_float3(-0.000334059457f,-0.00104828776f,

0.927569778f))

//define xyz_to_acesAP0 make_mat3(make_float3(1.0498110175f, 0.0f, -0.0001f), make_float3(-0.4959030231f,

1.3733130458f, 0.0982400361f), make_float3(0.0f, 0.0f, 0.9912520182f))

// OLog Curve Encoding Function

float relativeSceneLinearToNormalizedOLog( float x)

  {return (0.139f * log(x * 16+ 0.019f)/log(_expf(1)) + 0.614f);}

// OLog Curve Decoding Function

float normalizedOLogToRelativeSceneLinear( float x)

  {return (_expf((x - 0.614f)/0.139f) - 0.019f)/16.0f*7.37235f;;}


void OLogToACES

  ( input varying float rIn,

                input varying float gIn,

                input varying float bIn,

                input varying float aIn,

                output varying float rOut,

                output varying float gOut,

                output varying float bOut,

output varying float aOut)

{float r_lin = normalizedOLogToRelativeSceneLinear(rIn);

float g_lin = normalizedOLogToRelativeSceneLinear(gIn);

float b_lin = normalizedOLogToRelativeSceneLinear(bIn);

float x_D65 = r_lin * 0.6370 + g_lin * 0.1446 + b_lin * 0.1689;

float y_D65 = r_lin * 0.2627 + g_lin * 0.6780 + b_lin * 0.0593;

float z_D65 = r_lin * 0.0 + g_lin * 0.0281 + b_lin * 1.0610;
```

```
float x_D60 = x_D65 * 1.01174414 + y_D65 * 0.00770577991 + z_D65 * -0.0157216747;

float y_D60 = x_D65 * 0.00555788933 + y_D65 * 1.00153586 + z_D65 * -0.00626219941;

float z_D60 = x_D65 * -0.000334059457 + y_D65 * -0.00104828776 + z_D65 * 0.927569778;

rOut = x_D60 * 1.0498110175 + y_D60 * 0.0 + z_D60 * -0.0001;

gOut = x_D60 * -0.4959030231 + y_D60 * 1.3733130458 + z_D60 * 0.0982400361;

bOut = x_D60 * 0.0 + y_D60 * 0.0 + z_D60 * 0.9912520182;

aOut = 1.0;}
```