Practical HDR and Wide Color Techniques in Gran Turismo SPORT

SIGGRAPH ASIA 2018

Hajime UCHIMURA, Polyphony Digital Inc. / uchimura@polyphony.co.jp
Kentaro SUZUKI, Polyphony Digital Inc. / ksuzuki@polyphony.co.jp

Version 2018/12/22.

1. Introduction of This Course (5 min.)
2. Introduction of Color and HDR (25 min.)
3. Practical HDR Output Techniques (10 min.)
4. Asset Creation (15 min.)
5. Implementations and Results in Our Game (30 min.)
6. HDR Working Environment (10 min.)

Presenters

Hajime UCHIMURA

- Graphics programmer at Polyphony Digital Inc.
- Image processing and color science.

Kentaro SUZUKI

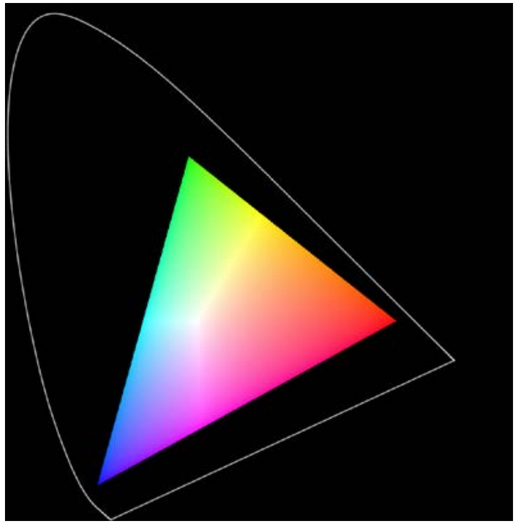
- Graphics programmer at Polyphony Digital Inc.
- Post effects and so on.

Gran Turismo SPORT

- Gran Turismo (GT)
  - “AAA” racing game series on PlayStation® platform
- Gran Turismo SPORT
  - Released in Oct. 2017
  - Highly reputation due to utilization of HDR and Wide Color technology.

Past of Real-time Rendering

- The quality of CG images has improved significantly over the past few years.
- However, final output quality was limited by conventional output devices.

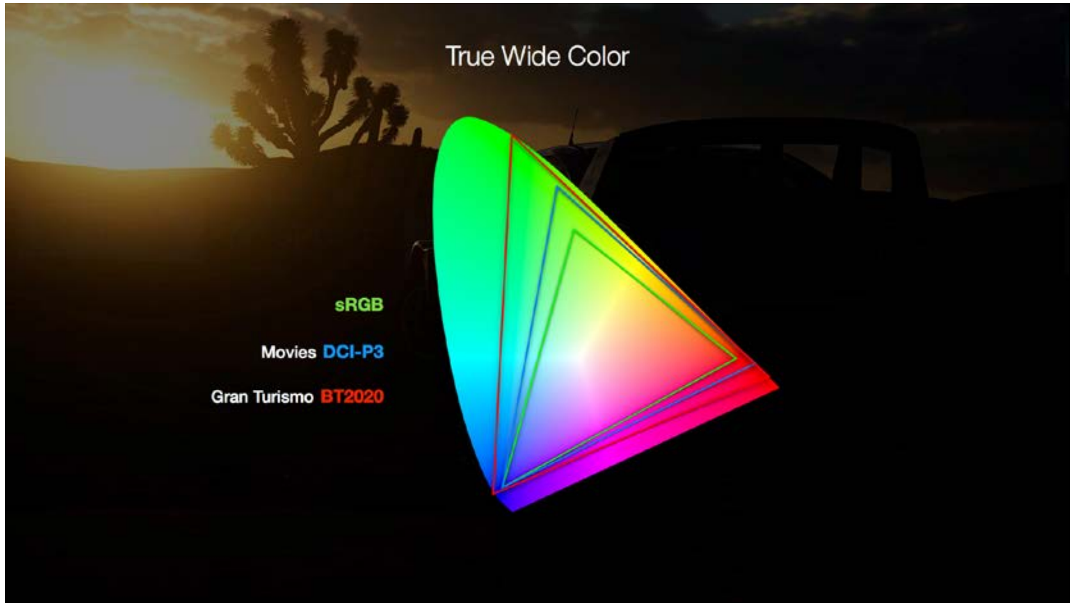


Limited Color Space

Limited Dynamic Range

New Technology: HDR and Wide Color

- HDR and Wide color technology has expanded these limitations.



Problems

- Finding a consistent interpretation between hardware behavior and software specification is difficult.
- As a result, implementing HDR and Wide Color technologies is difficult and non-trivial.

Our Solution from Game Industry

- Consistent theory-based approach for each aspect of the workflow
  - Asset collecting and editing
  - Interchangeable formats
  - Encoding
  - Working environment
  - Verification
  - Rendering pipeline
  - etc
- As a result, high quality output is achieved.

What You will Learn from This Course

Fundamental Theory

Practical Implementation

↓

You will acquire knowledge about developing HDR and wide color applications

1. Introduction of This Course (5 min.)
2. Introduction of Color and HDR (25 min.)
3. Practical HDR Output Techniques (10 min.)
4. Asset Creation (15 min.)
5. Implementations and Results In Our Game (30 min.)
6. HDR Working Environment (10 min.)

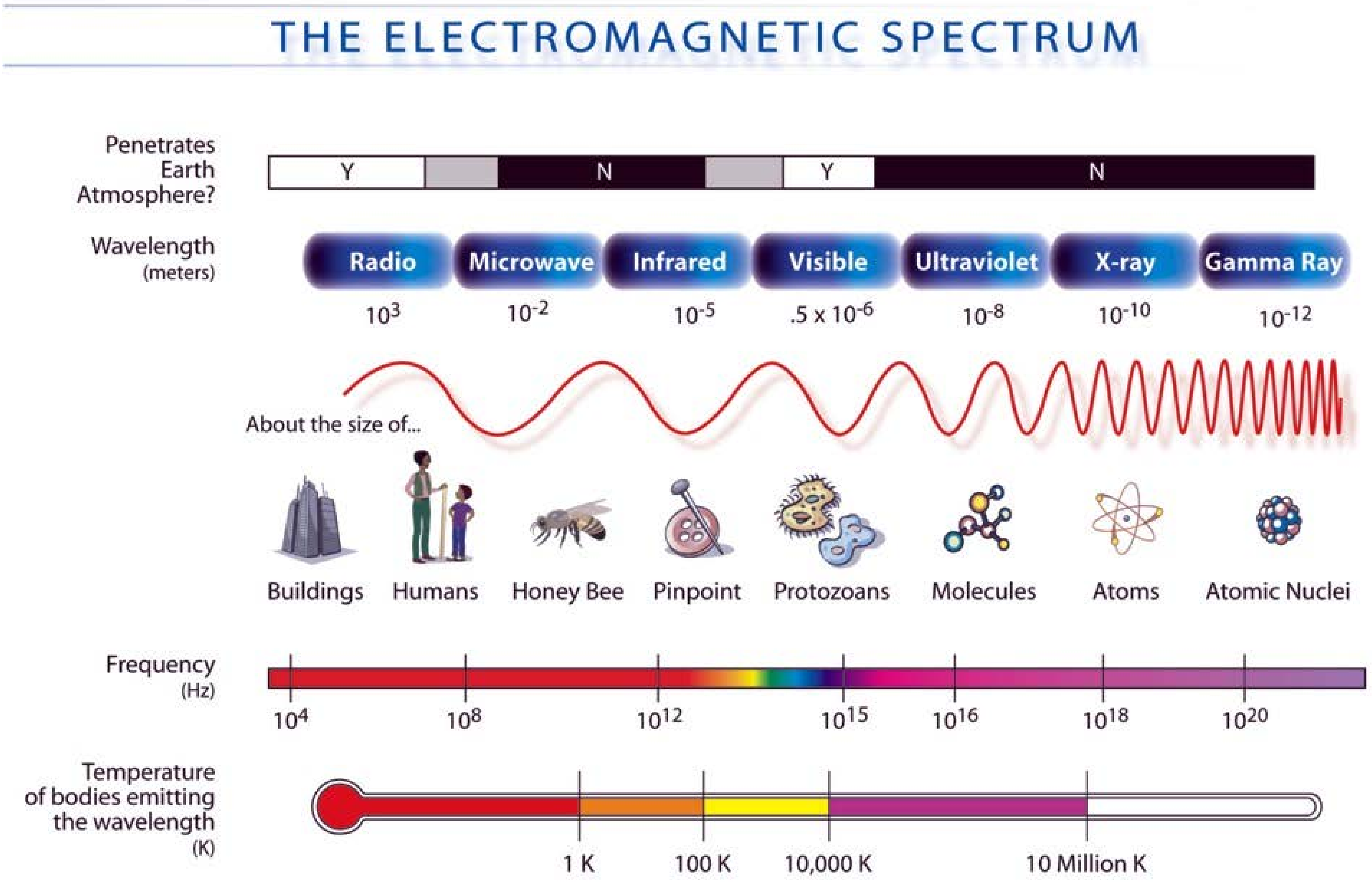
Course Materials

- http://www.polyphony.co.jp/publications/sa2018/

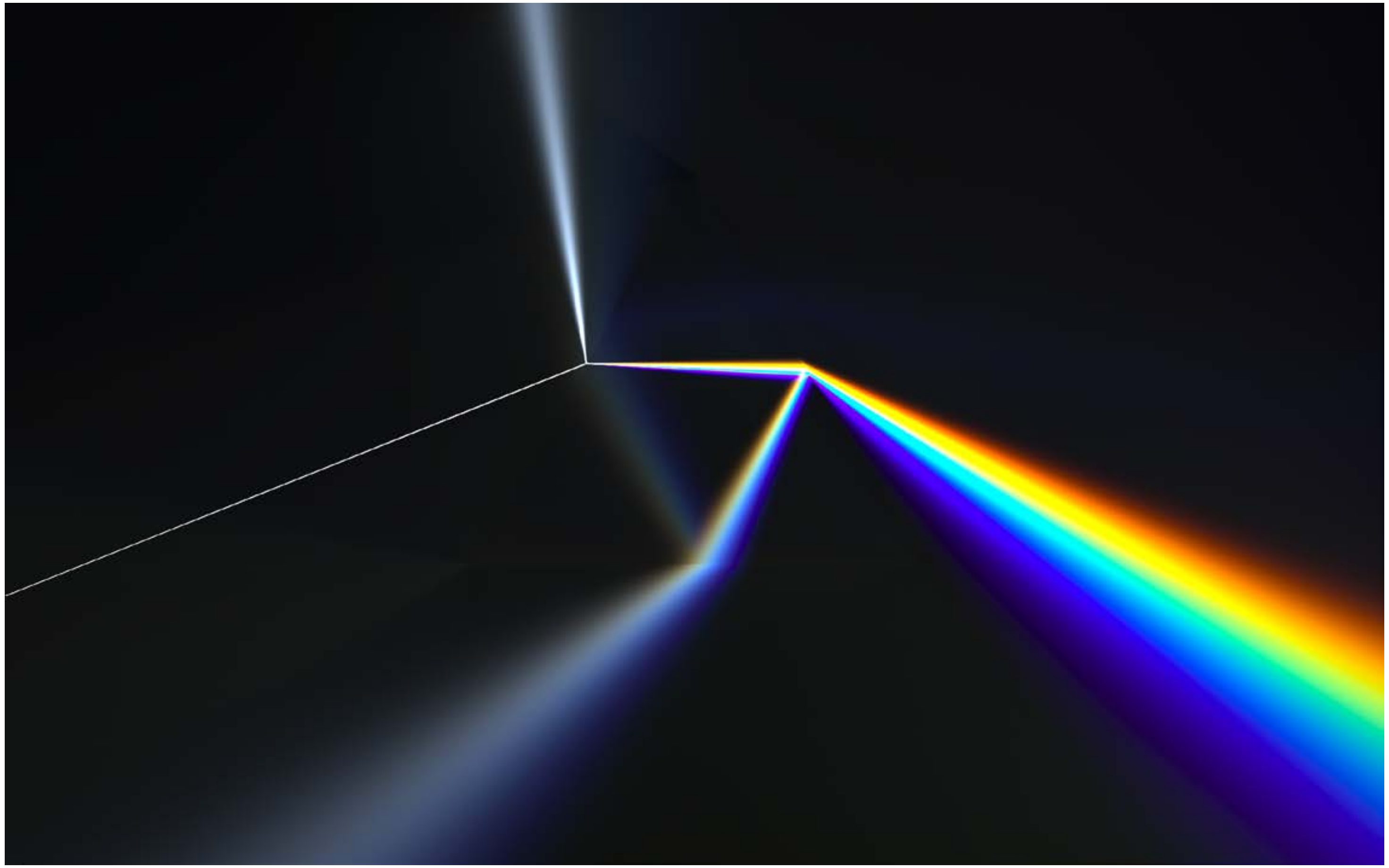
1. Introduction of This Course (5 min.)
2. Introduction of Color and HDR (25 min.)
3. Practical HDR Output Techniques (10 min.)
4. Asset Creation (15 min.)
5. Implementations and Results in Our Game (30 min.)
6. HDR Working Environment (10 min.)

1. Introduction of This Course (5 min.)
2. Introduction of Color and HDR (25 min.)
    a. Mechanism of Color Sensitivity
    b. High Dynamic Range Display
    c. Mechanism of Image transfer
    d. Wide Color Gamut
3. Practical HDR Output Techniques (10 min.)
4. Asset Creation (15 min.)
5. Implementations and Results in Our Game (30 min.)
6. HDR Working Environment (10 min.)

Eye sees Electro Magnetic Wave



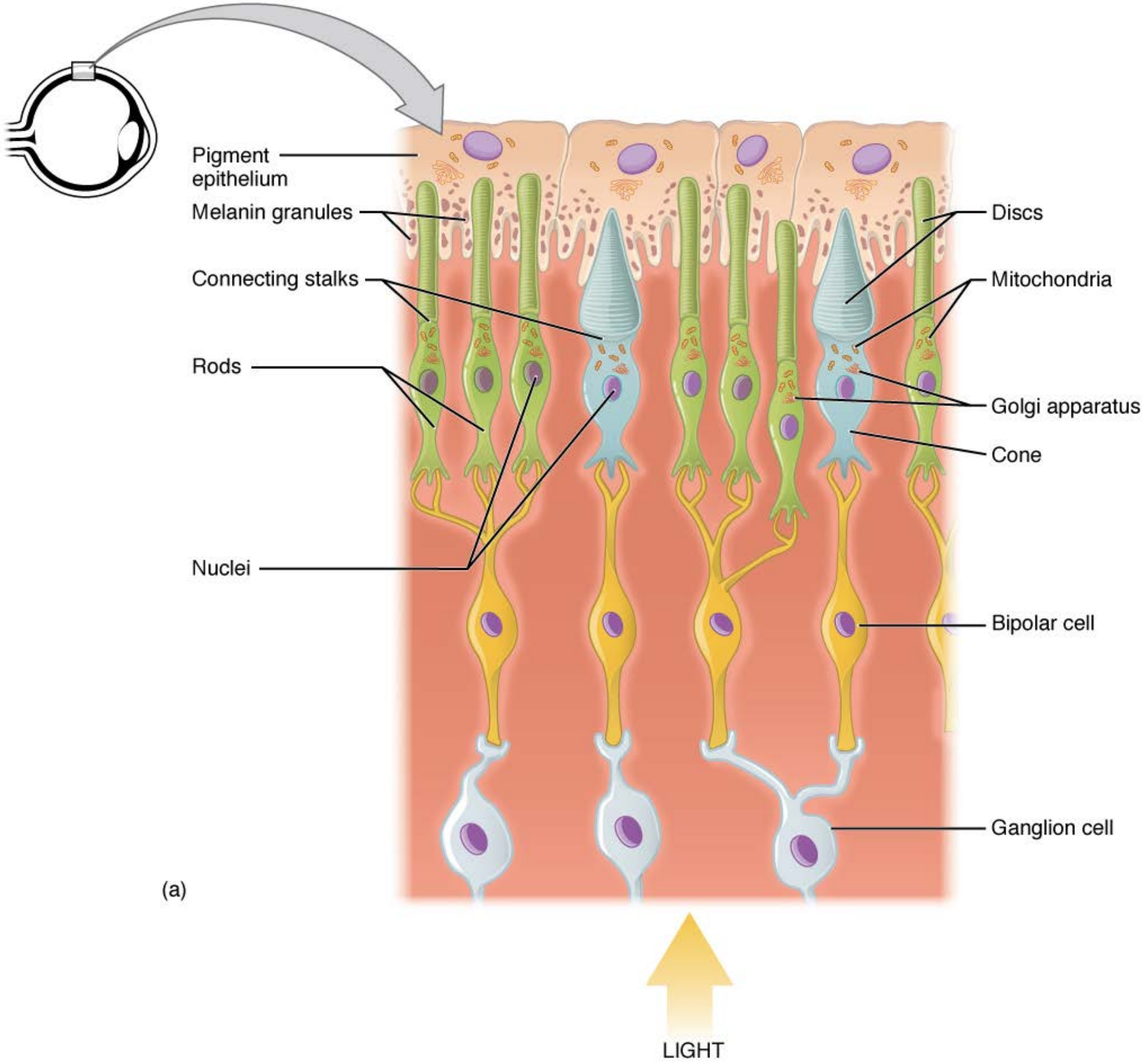
Light Consists of Spectrum



Spectral Color

Each Frequency of Spectrum has Unique Color Stimulus

Cones (and Rod)



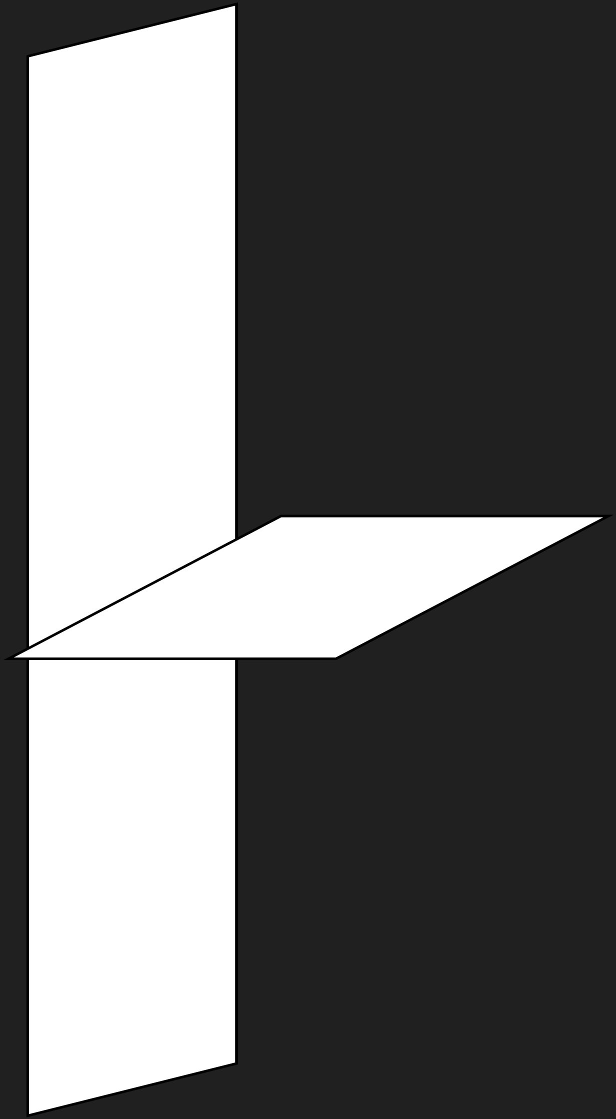
[College 2013]

Measuring Human Eye

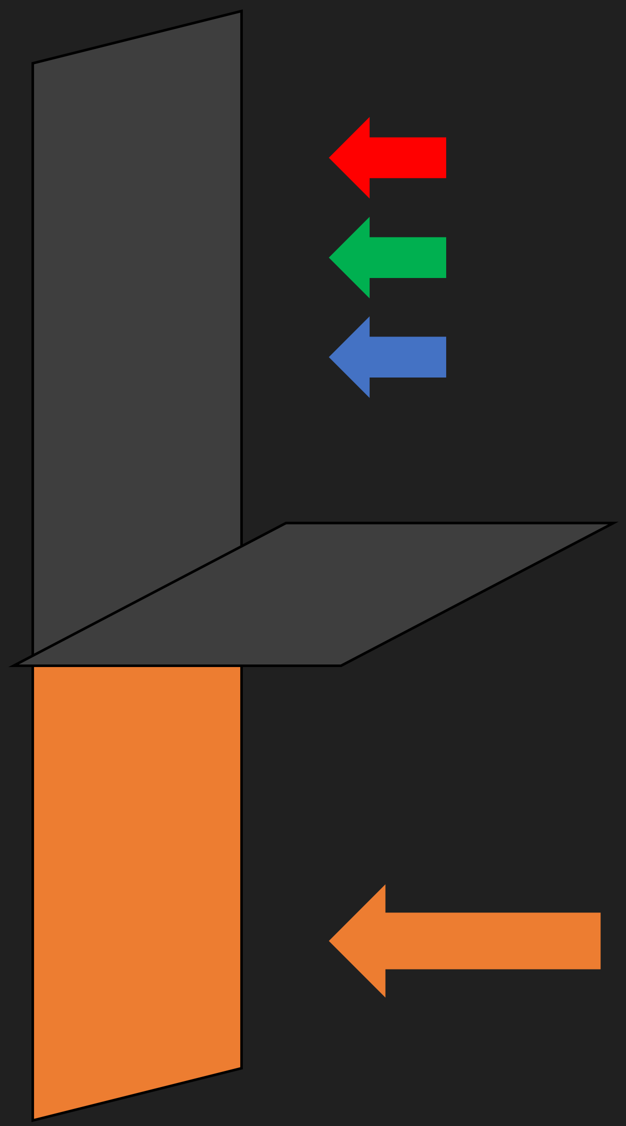
- CIE(Commission Internationale de l'Éclairage) did it in 1931. [Smith and Guild 1931]

http://www.cie.co.at/about-cie

Preparation



Light up Each Part



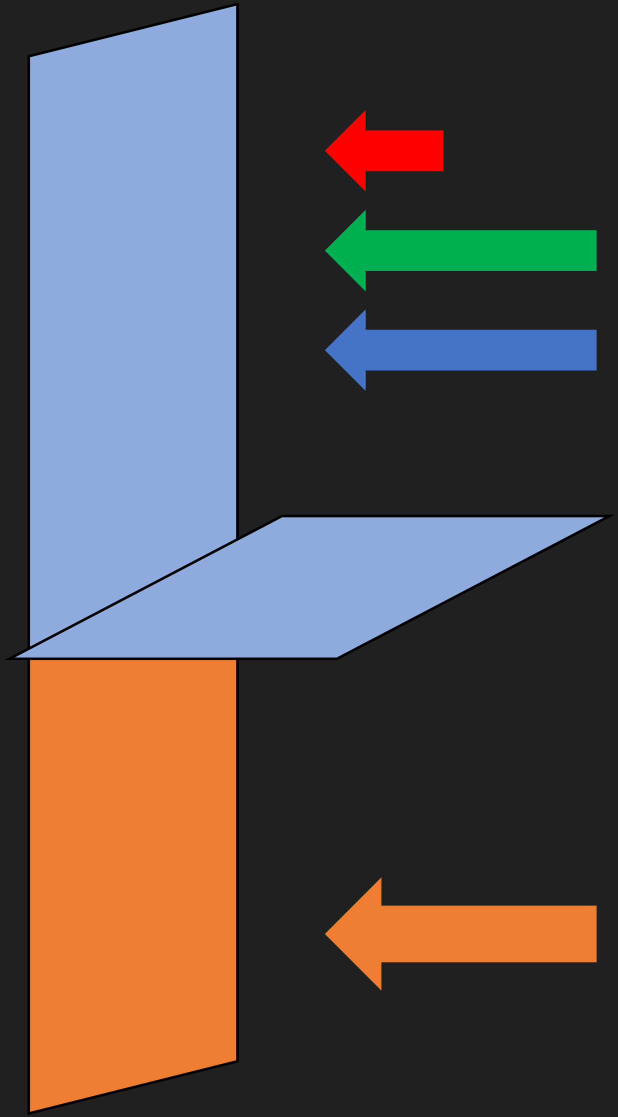
Red primary light

Green primary light

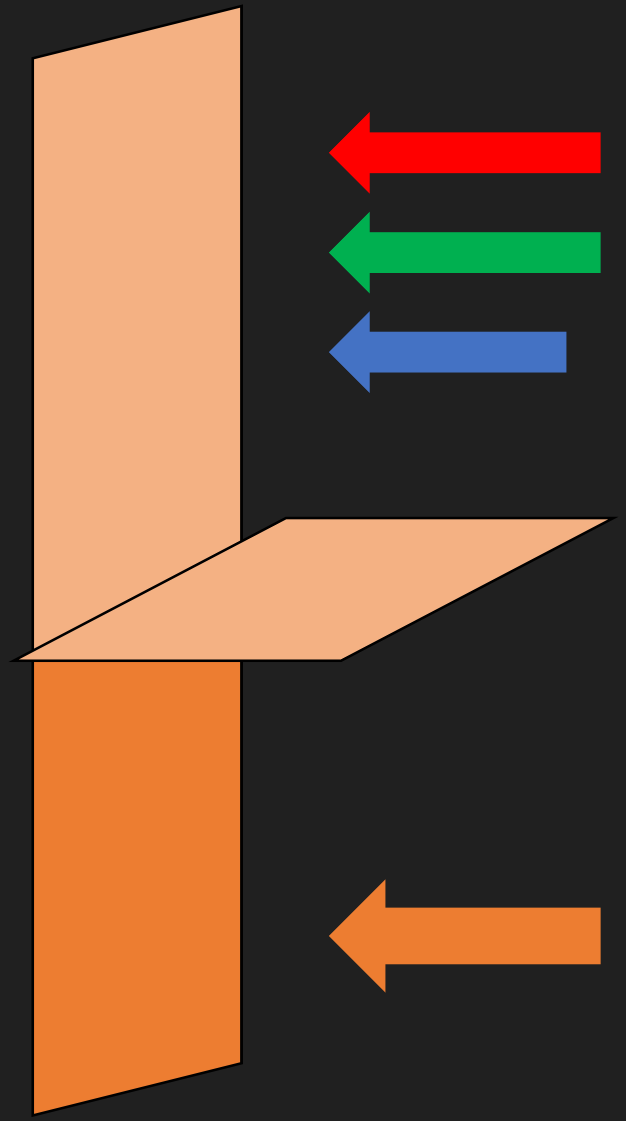
Blue primary light

Target light we want to measure.

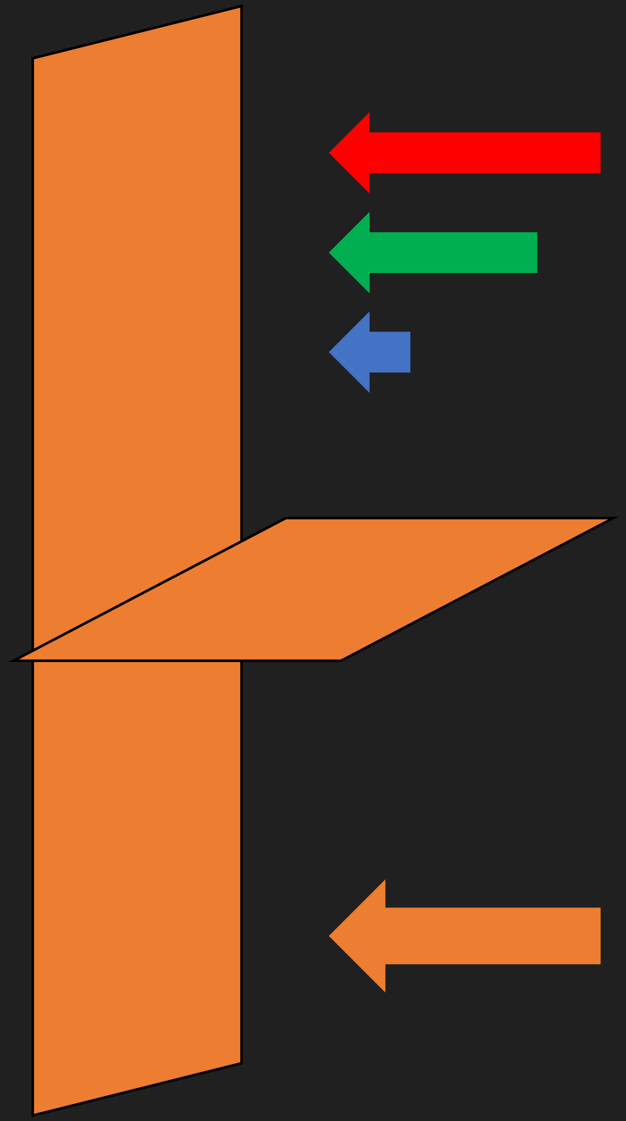
Adjust Color Balance



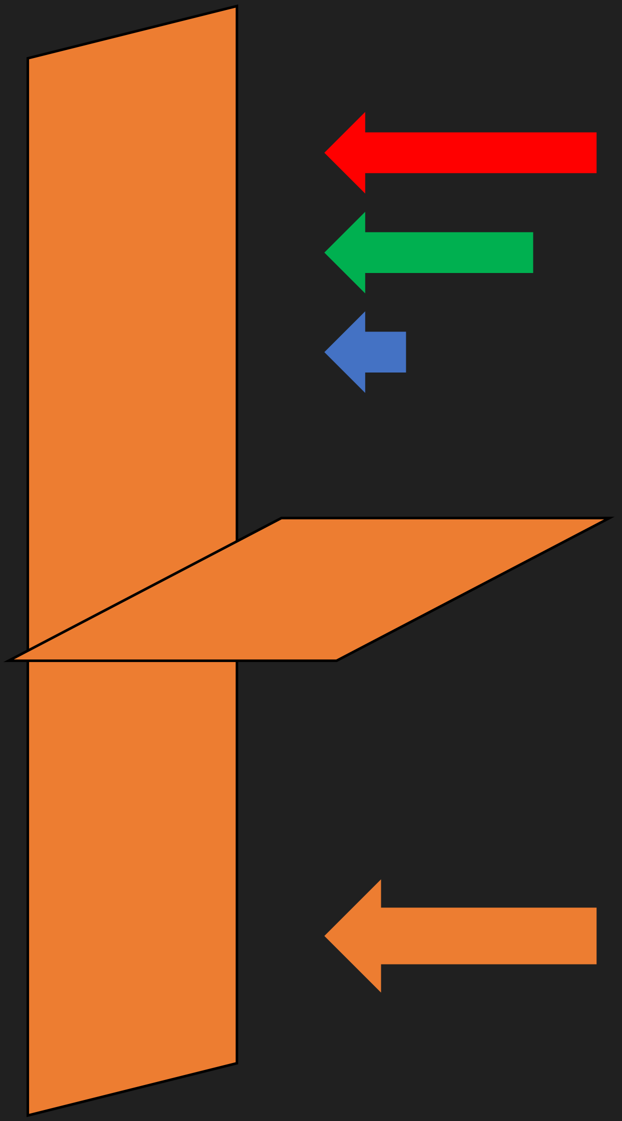
Adjust Color Balance



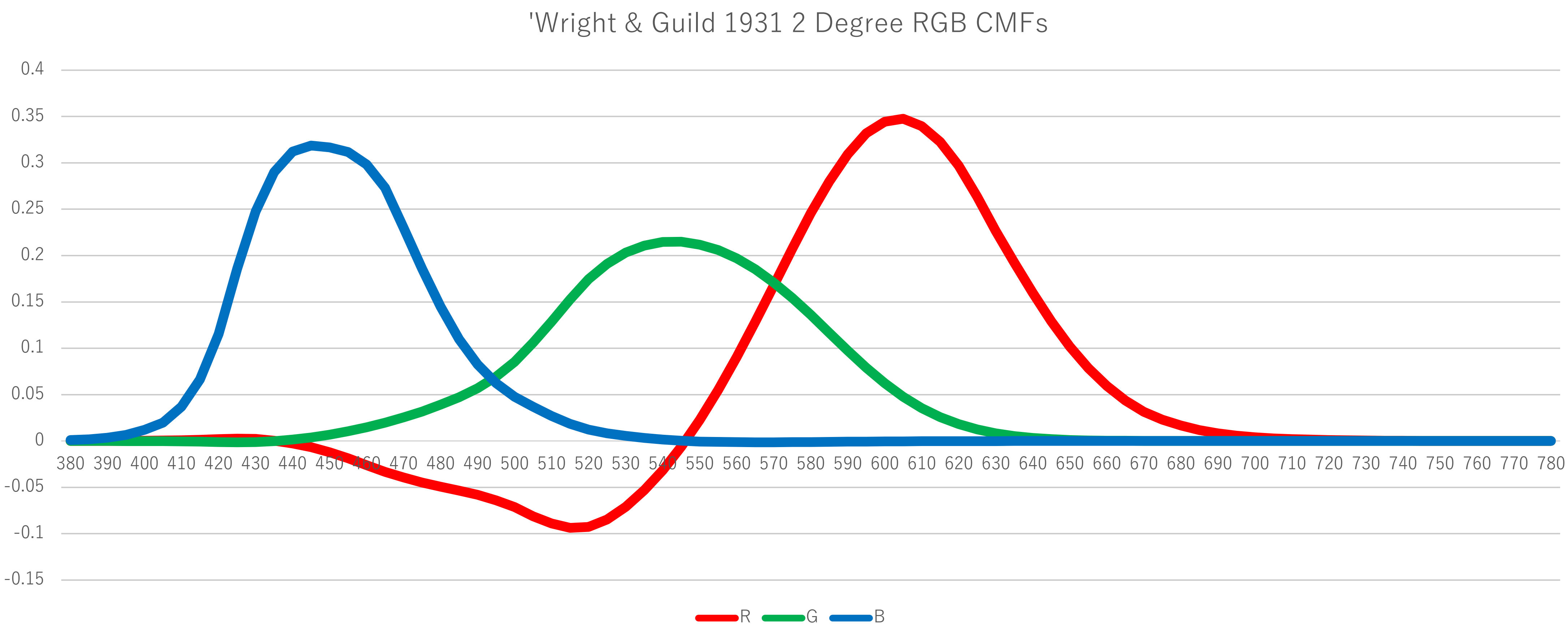
Adjust Until Color Matches



Repeat for All Visible Spectrum



Color Matching Functions (CMFs)



[Colour 2018]

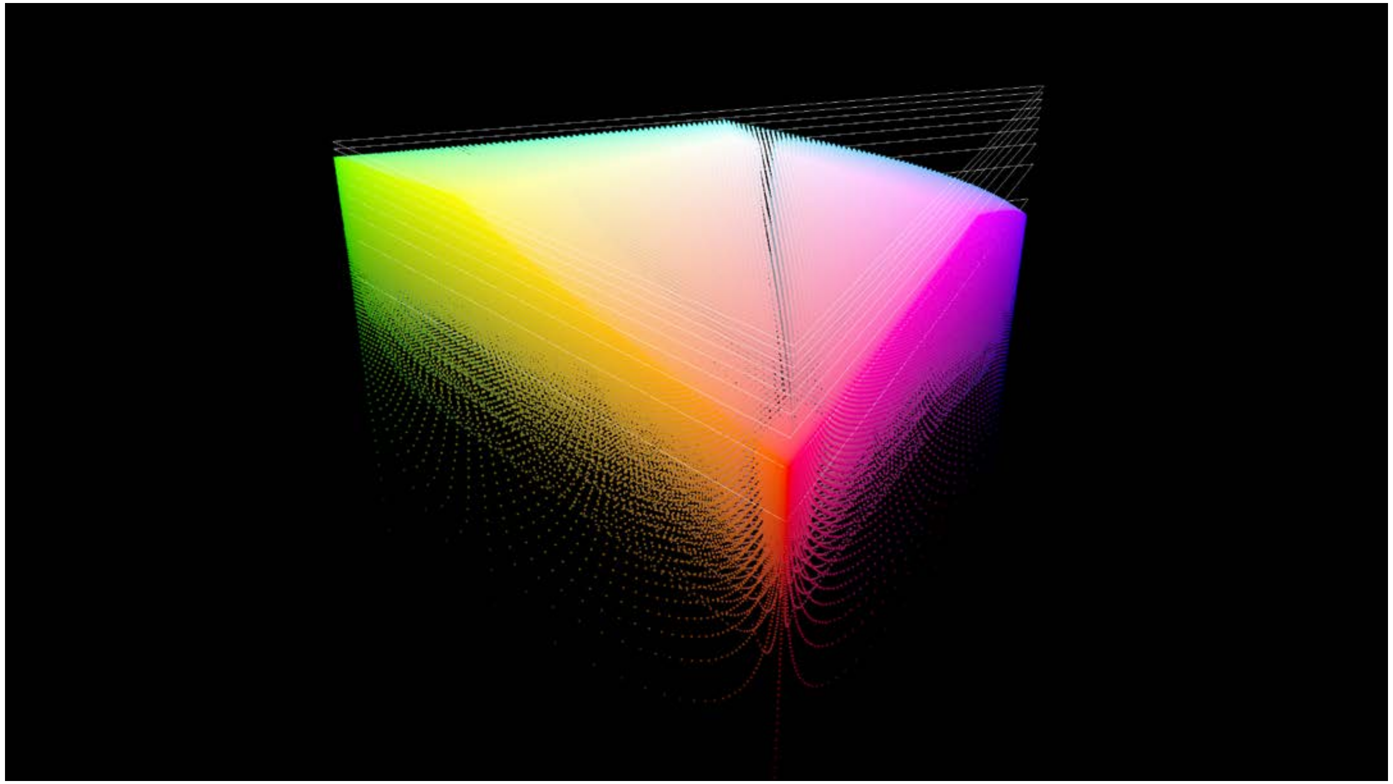
Variation of CMFs

- CIE 1931 XYZ
  - The standard.
- CIE 1931 XYZ Judd-Vos-modified (Judd Vos)
  - Better at blue color representation.
- CIE 2012 XYZ(CIE2006)
  - New CMFs

[Brindley 1955][CIE 1931][Judd 1951] [Vos 1978][Wyszecki and Stiles 1982][CIE 2006]
[Stockman et al. 1999][Stockman and Sharpe 2000][Sharpe et al. 2005][Sharpe et al. 2011]

Color Representation

- Three degrees of freedom is sufficient to describe color.
- This three dimensional vector is called “tristimulus value”.



Tristimulus Value Calculation

- Given a spectrum $C(\lambda)$ and CMFs$\left(\bar{x}(\lambda),\ \bar{y}(\lambda),\ \bar{z}(\lambda)\right)$, you get tristimulus (X,Y,Z) by integral below:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} \int C(\lambda)\bar{x}(\lambda)d\lambda \\ \int C(\lambda)\bar{y}(\lambda)d\lambda \\ \int C(\lambda)\bar{z}(\lambda)d\lambda \end{pmatrix}$$

xy Coordinates

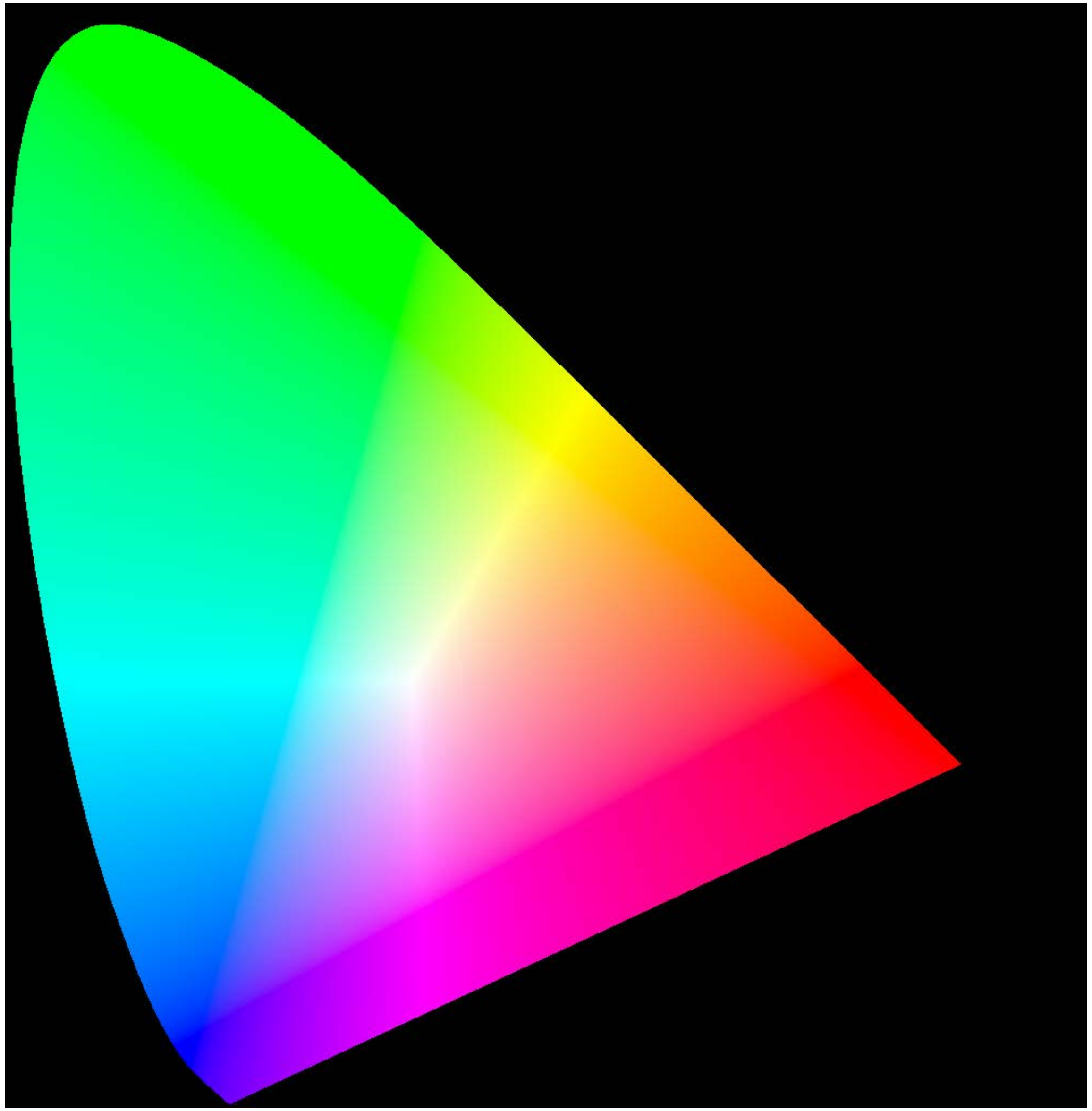
- We calculate xy coordinate (x, y) from tristimulus value $(\mathrm{X}, \mathrm{Y}, \mathrm{Z})$ as below:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \dfrac{X}{X+Y+Z} \\ \dfrac{Y}{X+Y+Z} \end{pmatrix}$$

- This xy coordinate is often used when talking about any light's absolute color.

Color Locus

- “Chromaticity diagram”.
- All visible colors.
- “Color locus”
- No one can see colors outside. “Imaginary color”.



Luminance Unit

- $\bar{y}(\lambda)$ of CIE 1931 XYZ CMFs is the exactly same function of Luminosity function of human eye.
- Luminous intensity in candela of any light wave $I_e(\lambda)$ can be calculated as this formula:

$$I_v(\lambda)_{cd} = 683.002\frac{\mathrm{lm}}{W} \cdot \bar{y}(\lambda) \cdot I_e(\lambda)_{\frac{W}{Sr}}$$

- $1\ cd/m^2 = 1\ nit$.

Color Matching

- Given tristimulus values $C_1 = (x_1, y_1, z_1)$ and $C_2 = (x_2, y_2, z_2)$

$C_1 = C_2$ means those colors are the same.

- Color matches even if spectrum is different.
- “metamerism”

Tristimulus Color Basic Rules

- Rule 1:

$$\boldsymbol{C}_1 = \boldsymbol{C}_2 \text{ then } K\boldsymbol{C}_1 = K\boldsymbol{C}_2$$

- Rule 2:

$$\text{When } \boldsymbol{C}_1 = \boldsymbol{C}_2 \; and \; \boldsymbol{C}_3 = \boldsymbol{C}_4$$
$$then \; \boldsymbol{C}_1 + \boldsymbol{C}_3 = \boldsymbol{C}_2 + \boldsymbol{C}_4$$

- These rules are established from the experience. [Grassman 1953]

Tristimulus Color Interpolation

- Color can be interpolated in form of tristimulus value.
- Given color C_1 and C_2 , a color between those two can calculated as

$$aC_1 + \ (1 - a)C_2$$

where a is a mixture ratio.

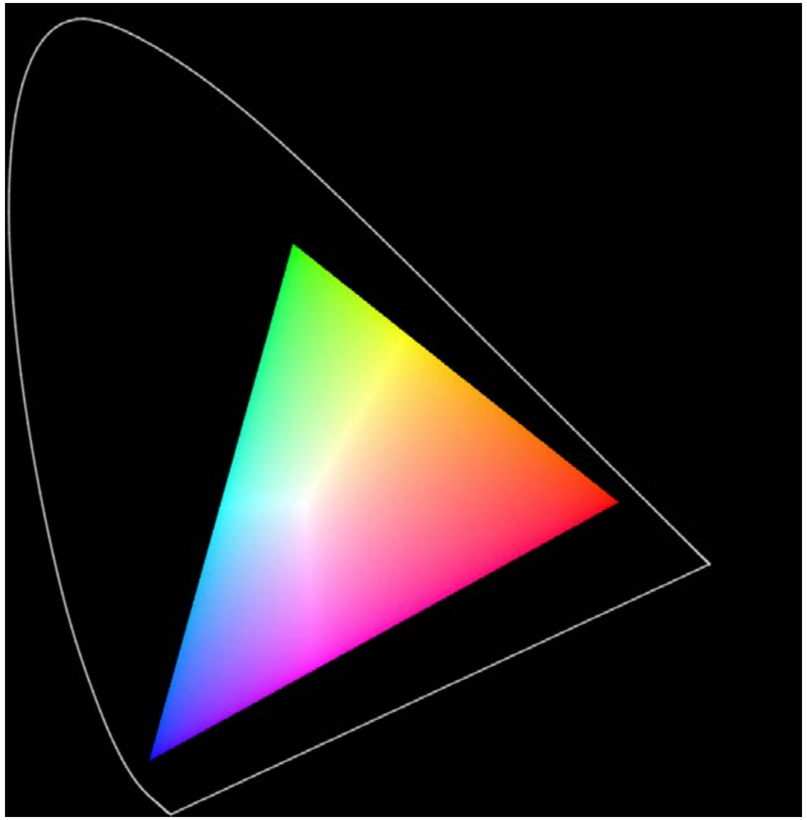
Three Primary Colors

- Select arbitrarily primary colors P_r, P_g, P_b
- These primary colors can describe another color

$$C(r, g, b) = rP_r + gP_g + bP_b$$

The Color Inside Primary Colors

$$C(r,g,b) = rP_r + gP_g + bP_b$$

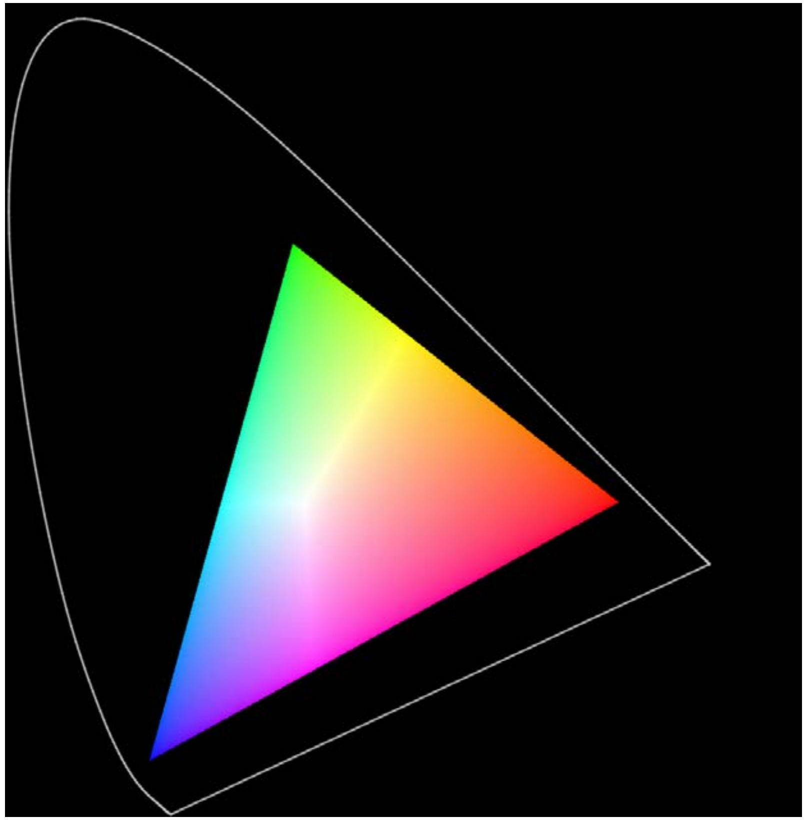


Gamut

- Colors inside three primary colors.

$$C(r, g, b) = rP_r + gP_g + bP_b$$

while $r > 0 \cap g > 0 \cap b > 0$



Gamut and Tristimulus

- Tristimulus value is colorimetric system dependent.
- “XYZ tristimulus”: CIE1931XYZ?

Gamut Conversion Matrix

- With different color primaries $(P_r, P_g, P_b)\ and\ (Q_r, Q_g, Q_b)$, the same color results in different tristimulus values.

$$rP_r + gP_g + bP_b = xQ_r + yQ_g + zQ_b$$

- So the conversion between two gamuts can be treated as a simple matrix transformation.

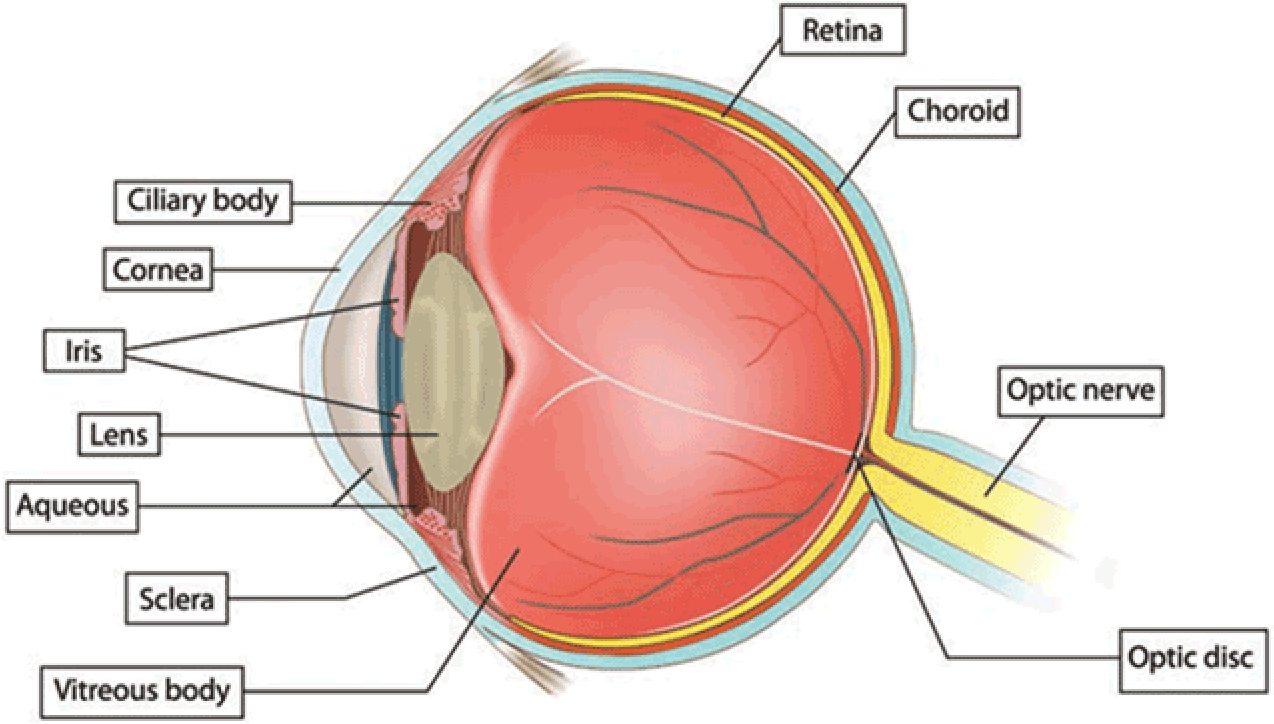
$$(r, g, b) = \begin{bmatrix} M_{11} & \cdots & M_{13} \\ \vdots & \ddots & \vdots \\ M_{31} & \cdots & M_{33} \end{bmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

Section conclusion

- Light is electromagnetic wave spectrum.
- Human eye perceives spectrum as a color.
- Color can be represented by a three dimensional vector.

1. Introduction of This Course (5 min.)
2. Introduction of Color and HDR (25 min.)
    a. Mechanism of Color Sensitivity
    b. High Dynamic Range Display
    c. Mechanism of Image Transfer
    d. Wide Color Gamut
3. Practical HDR Output Techniques (10 min.)
4. Asset Creation (15 min.)
5. Implementations and Results in Our Game (30 min.)
6. HDR Working Environment (10 min.)

Human Eye is Good Sensor

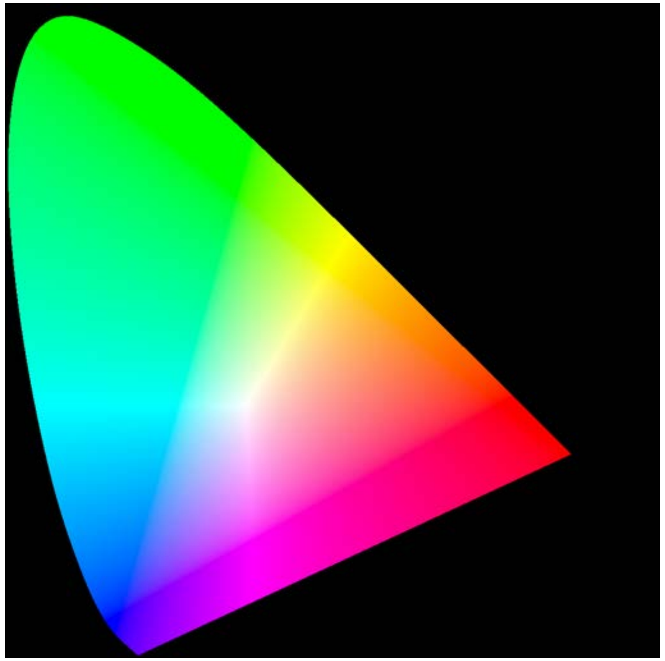


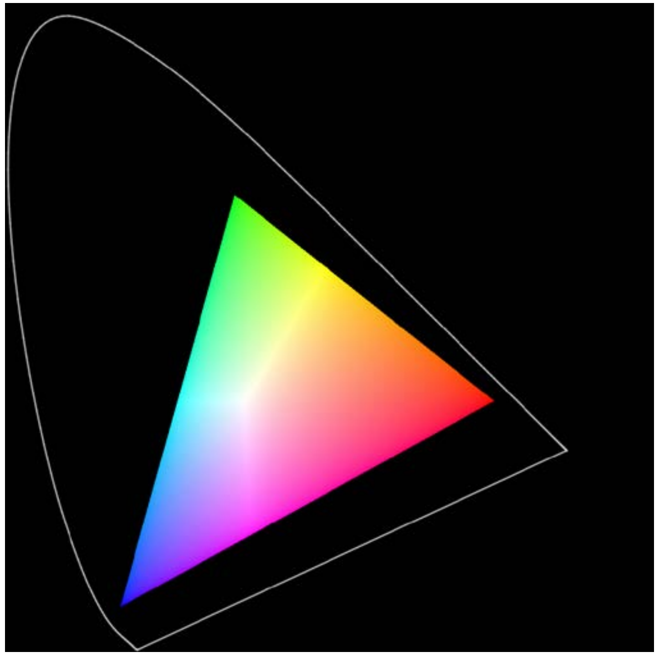
Dynamic Range of Human Eye

- Human eye can see from $10^{-6} cd/m^2$ to $10^8 cd/m^2$ with pupil adaptation. [Pirenne et al. 1957][Hallett 1969]
- Virtually human eye has 10^{14} dynamic range of luminance.
- The Dolby Imaging research team found 1:10000 of dynamic range in picture satisfied 90% of subjects. [Dolby 2014]

Color Range of Human Eye

- Human eye can see 1nm difference[Hatada 1985].
- The conventional TV covers less than 80%.





Display Nowadays

- **HDR**
  - Latest monitors.
  - Wider gamut.
  - Brighter peak luminance.
  - "HDR10"
- **SDR**
  - Conventional monitors.
  - Limited gamut.
  - Lower peak luminance.

Display Nowadays

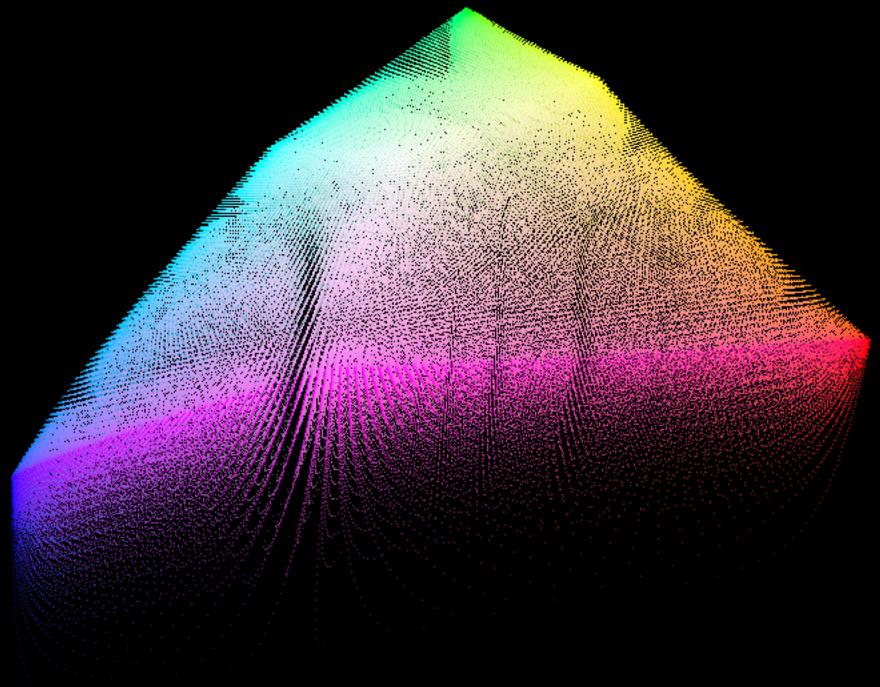
- **HDR**
  - Latest monitors.
  - Wider gamut.
  - Brighter peak luminance.
  - "HDR10"
- **SDR**
  - Conventional monitors.
  - Limited gamut.
  - Lower peak luminance.

LDR: Low Dynamic Range

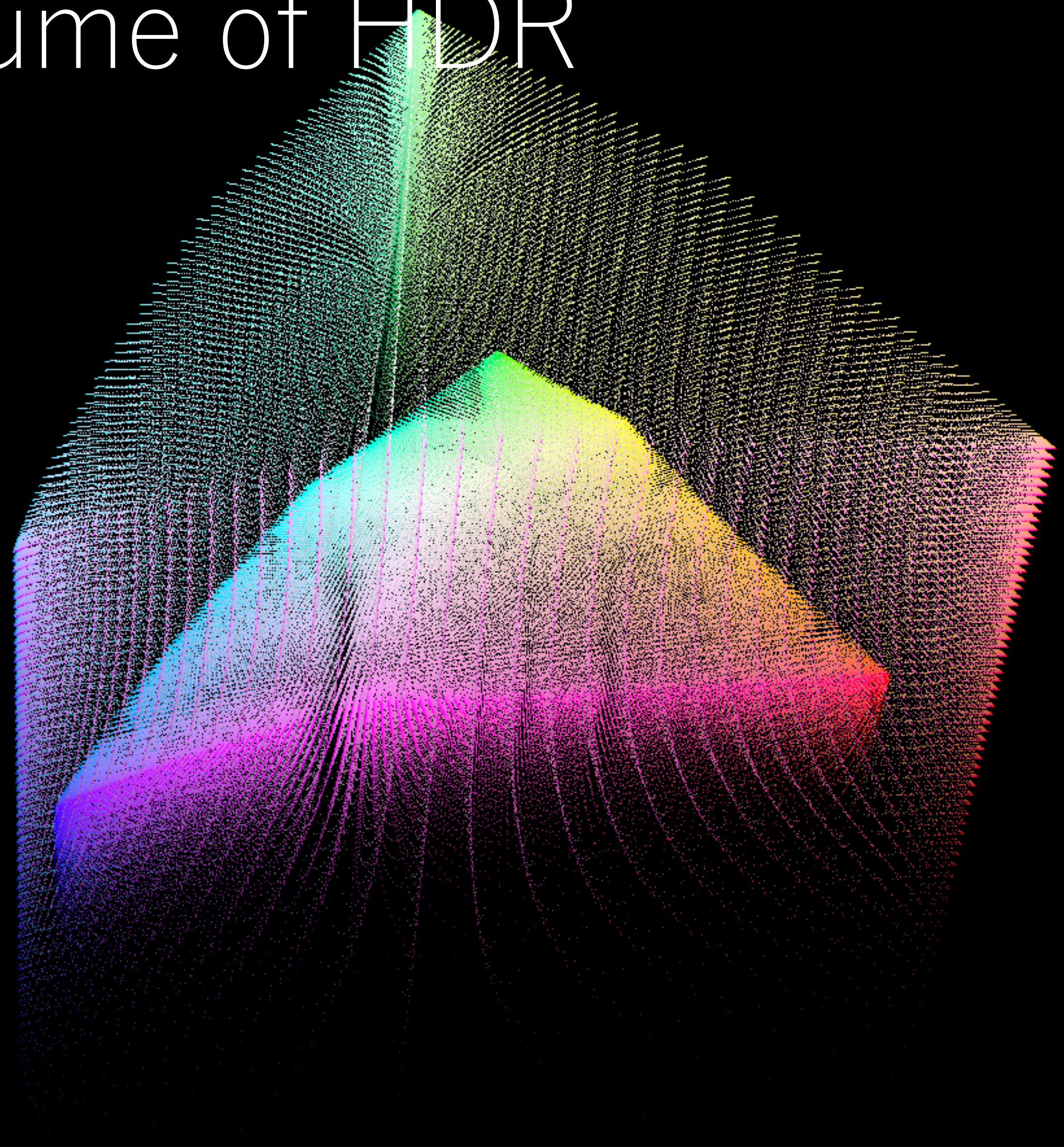
- Popular in Gaming industry.
- Conventional 0-255, 0-1 value.

- HDR means value which can be 1+.
  - float32, fp16, etc…

Color Volume of SDR



Color Volume of HDR



HDR TV: Complicated Market

- HDR TV varies from cheap to expensive.
  - Some TV that is just SDR TV with a bit modified gamma.
  - Some TV that is better than professional monitor.

- Device specification calibration is important.

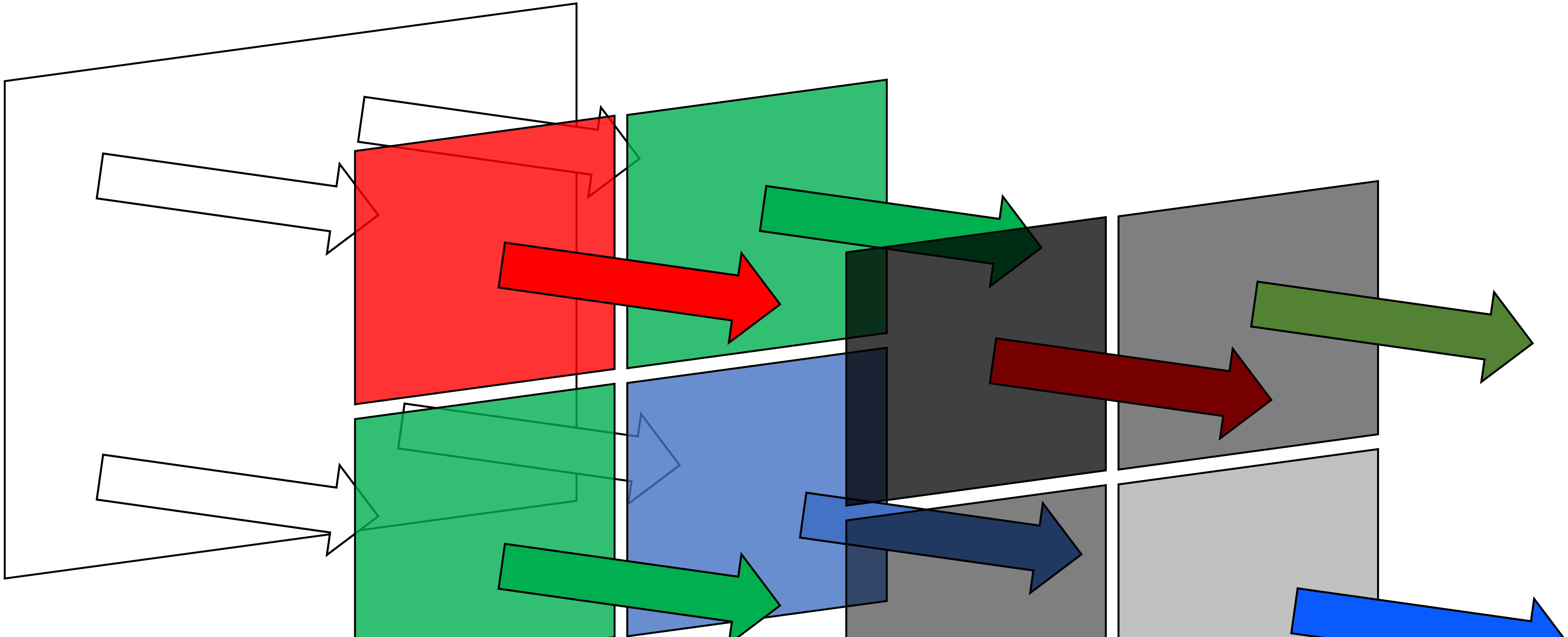

emilfreyracing.com
MICHELIN
new balance
Allianz
OMP
new balance

HDR TV Variations

- **Non emissive display**
  - LCD with backlight
  - Local dimming
- **Self emissive display**
  - OLED
  - Micro LED array

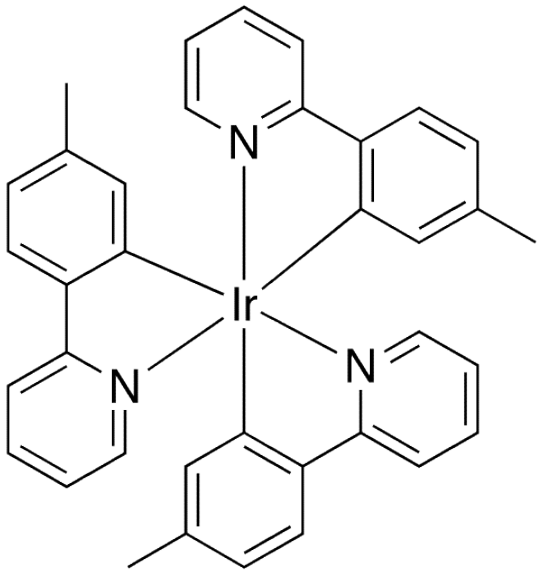
Non Emissive Display

- Basically LCDs. (Liquid Crystal Display)
- Backlight + color filtered active shutter in front.



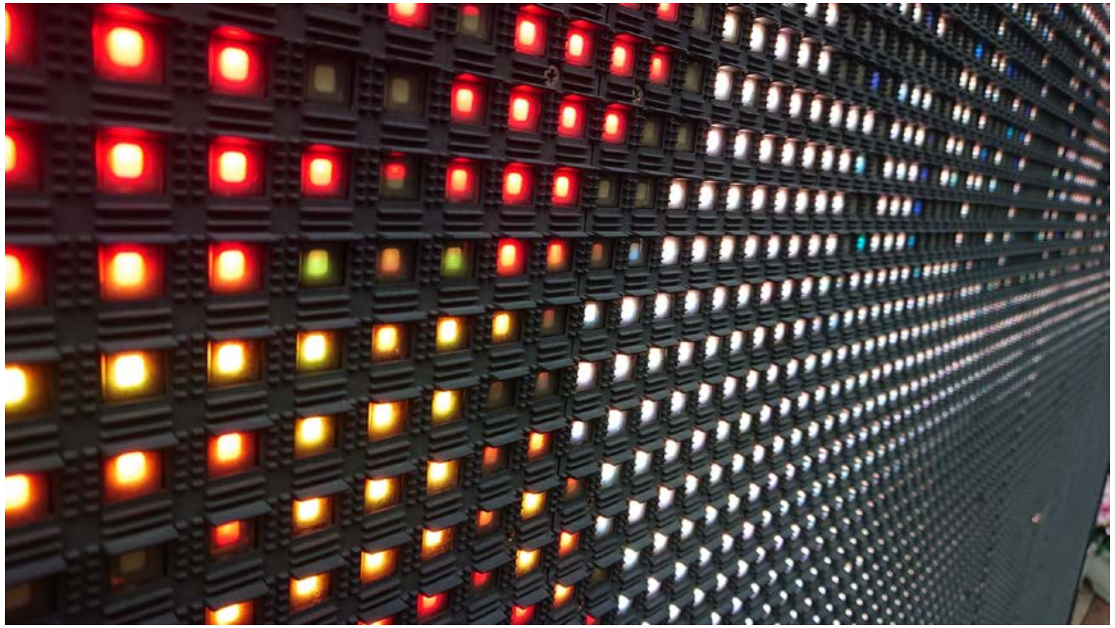
Self Emissive Displays

- OLED(Organic Light Emitting Device)
  - Good picture quality.
  - Perfect dark pixel.
  - Causes burn-in and damaged when picture is not moving.
  - Future games must think about it on OLED.



Self Emissive Displays

- Micro LED
  - Great picture quality.
  - No burn-in.
  - Very expensive.



1. Introduction of This Course (5 min.)
2. Introduction of Color and HDR (25 min.)
   a. Mechanism of Color Sensitivity
   b. High Dynamic Range Display
   c. Mechanism of Image Transfer
   d. Wide Color Gamut
3. Practical HDR Output Techniques (10 min.)
4. Asset Creation (15 min.)
5. Implementations and Results in Our Game (30 min.)
6. HDR Working Environment (10 min.)

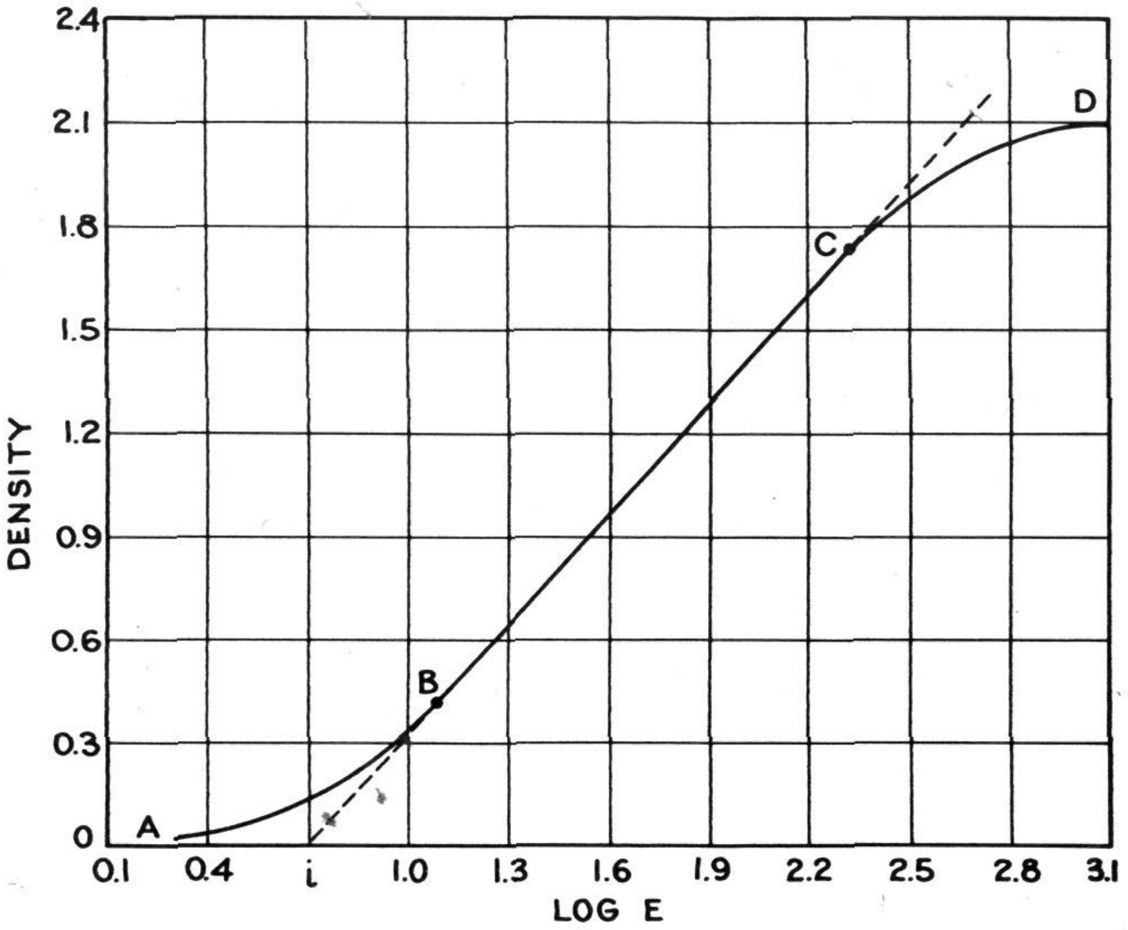
Human Eye Nonlinearity

- Human eye has nonlinear sensitivity to images.
- Eyes are very sensitive at dark parts, but are not as sensitive at brighter parts.

- Using nonlinear function can reduce bandwidth.

The Origin of Gamma Curve

- Photographic film. [Kenneth Mees 1954].
- Film tone reproduction characteristics was nonlinear log curve.
- This fits human eye sensitivity.



CHARACTERISTIC CURVE shows density versus exposure.

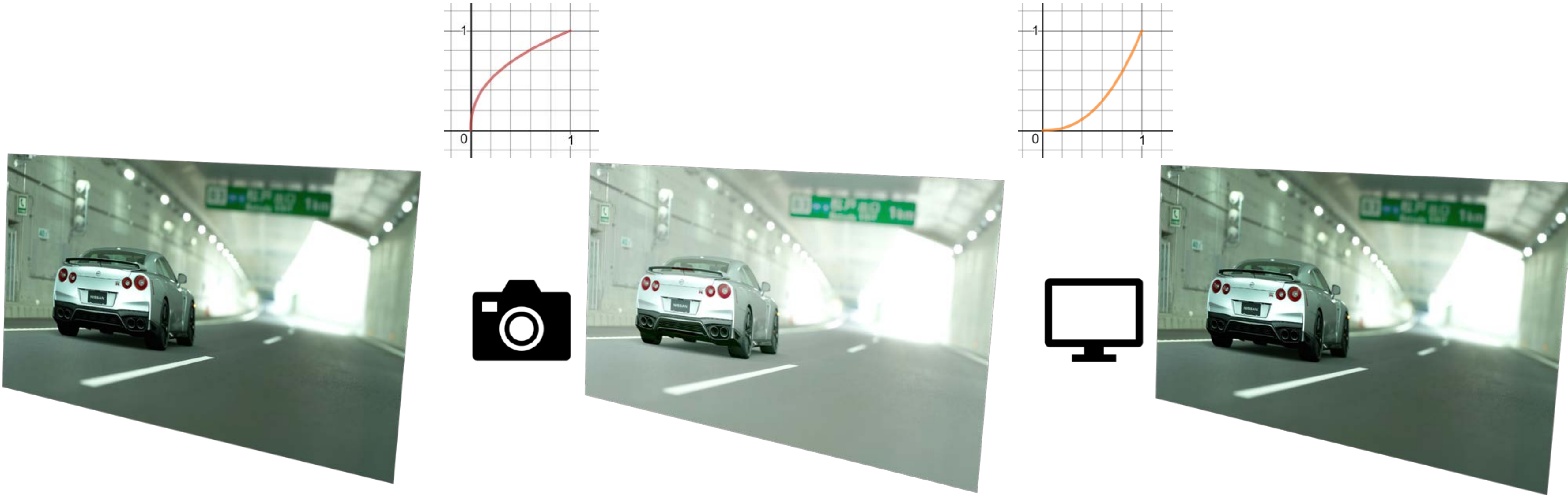
Gamma Curve in TV Monitor

- Reduces bandwidth of radio wave due to the same reason.
- Display tube nonlinearity.
- Keeps system simple and cheap.

OETF / EOTF

- **OETF**
  - Optical to Electric
  - Used in Camera.
- **EOTF**
  - Electric to Optical
  - Used in TV monitor.

OETF/EOTF



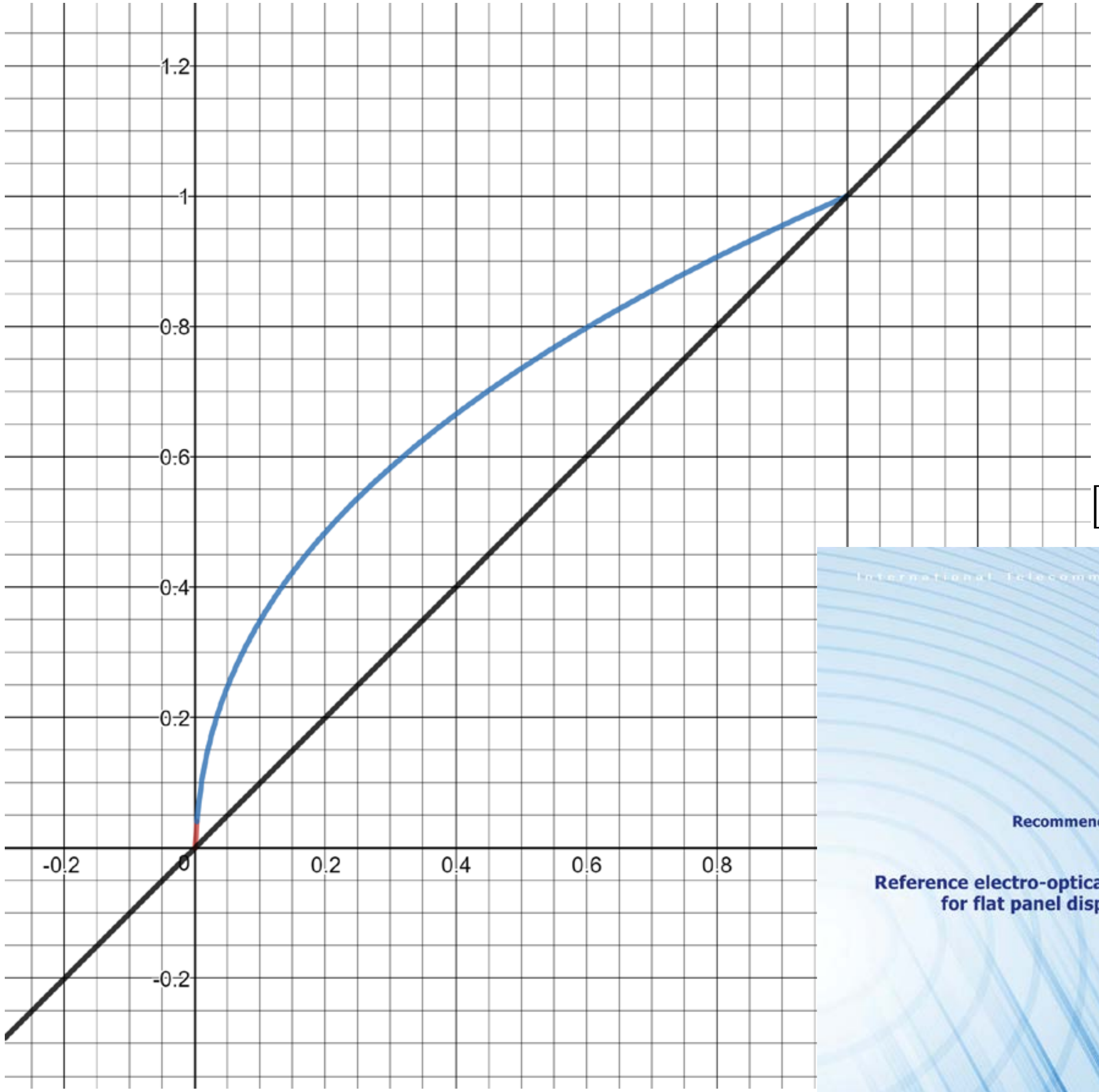
Linear optical

OETF applied image

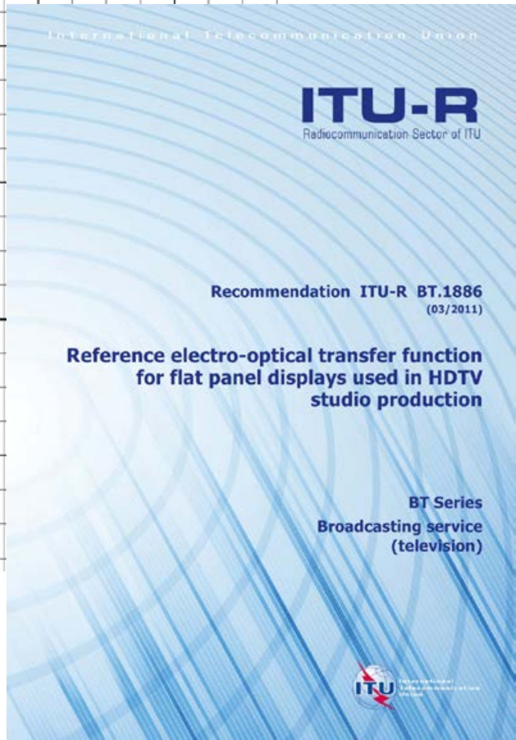
Output linear optical

sRGB Gamma (OETF)

- De facto standard.
- Almost all of SDR TV and PC monitors use this.
- Check ITU-R BT.1886 for more detail.



[BT1886]

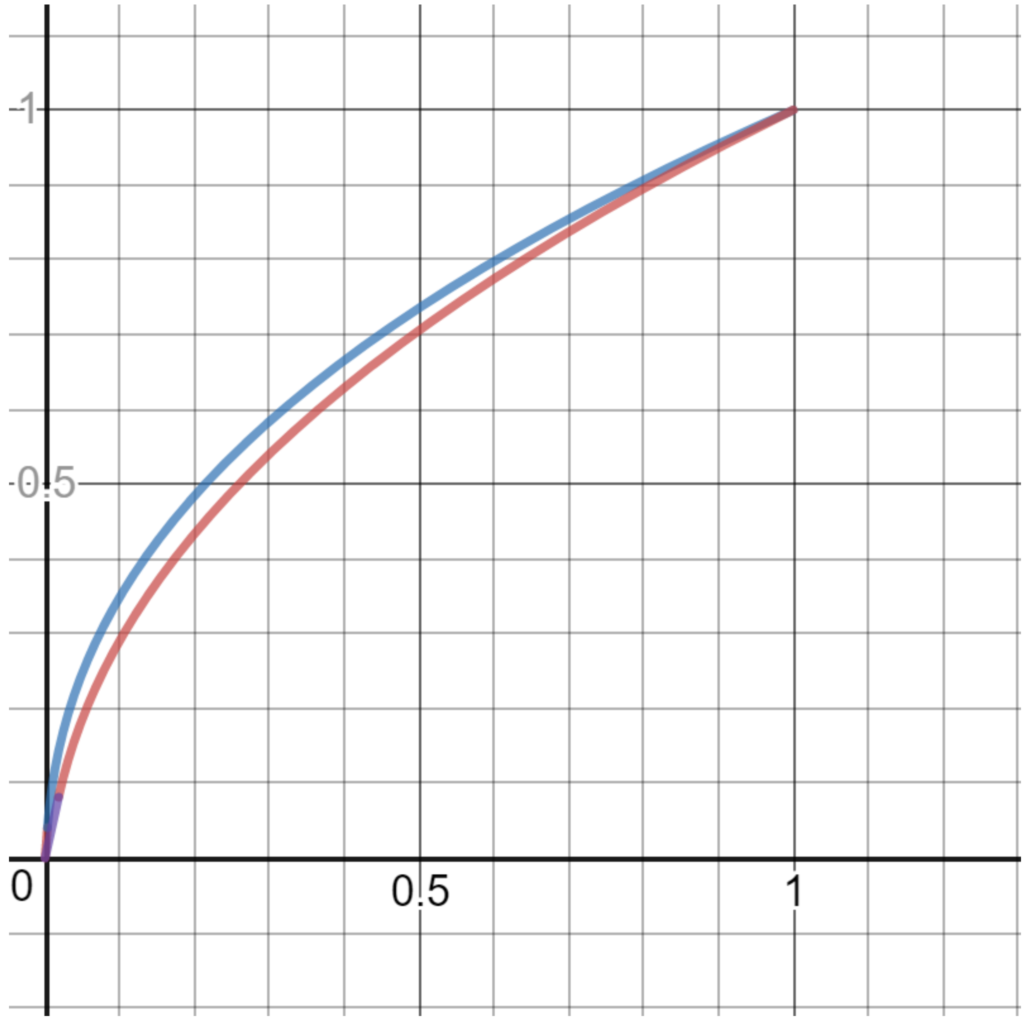


SDR Gamma

- Two variations are present.
- “ITU-R Recommendation BT.709” = Rec.709 gamma = BT.1886 gamma
  - $V = \begin{cases} 1.099L^{0.45} - 0.099 & (0.018 \leq x \leq 1) \\ 4.500L & (0 \leq x \leq 0.018) \end{cases}$
- sRGB gamma
  - $V = \begin{cases} 1.055L^{\frac{1}{2.2}} - 0.055 & (0.0031308 \leq x \leq 1) \\ 12.92L & (0 \leq x \leq 0.0031308) \end{cases}$

Complicated SDR Gamma

- Those two are slightly differ.
- Actual gamma may vary.
- We will use "sRGB gamma".



sRGB Gamut

- “ITU-R Recommendation BT.709” = Rec.709 = BT.1886 = sRGB
  - Different names, same gamut.
  - sRGB 100 nit is our standard of SDR.

	x	y
White	0.3127	0.3290
Red	0.64	0.33
Green	0.30	0.60
Blue	0.15	0.06

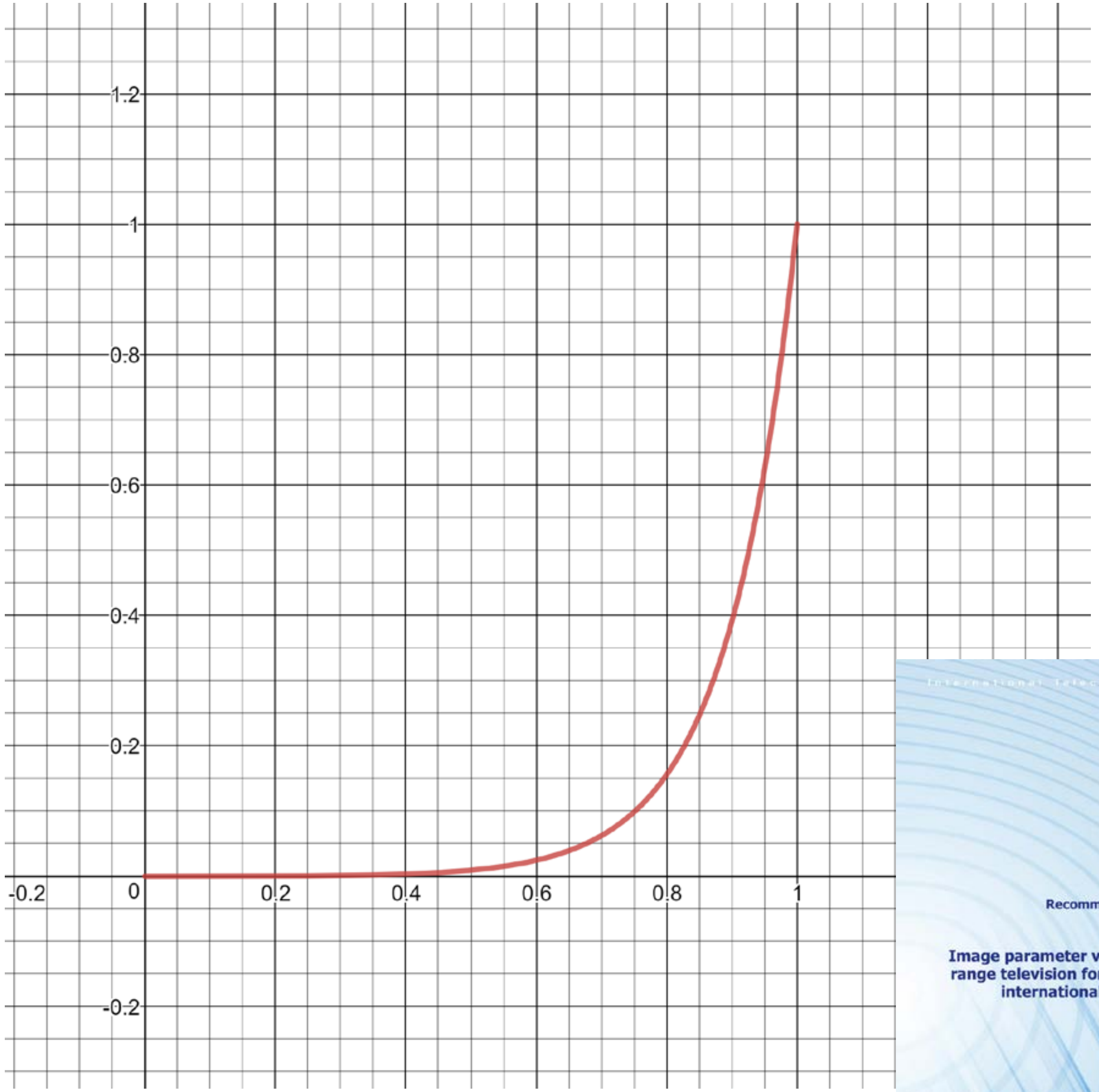
OETF/EOTF in HDR TV

- HDR10, PQ, Dolby, ITU-T BT.2100
  - Same Curve, different name.
  - Peak brightness 10,000 nits.
  - Absolute luminance.

- Hybrid Log-Gamma, HLG
  - Has compatibility with SDR TV.
  - Relative luminance.

PQ Curve (EOTF)

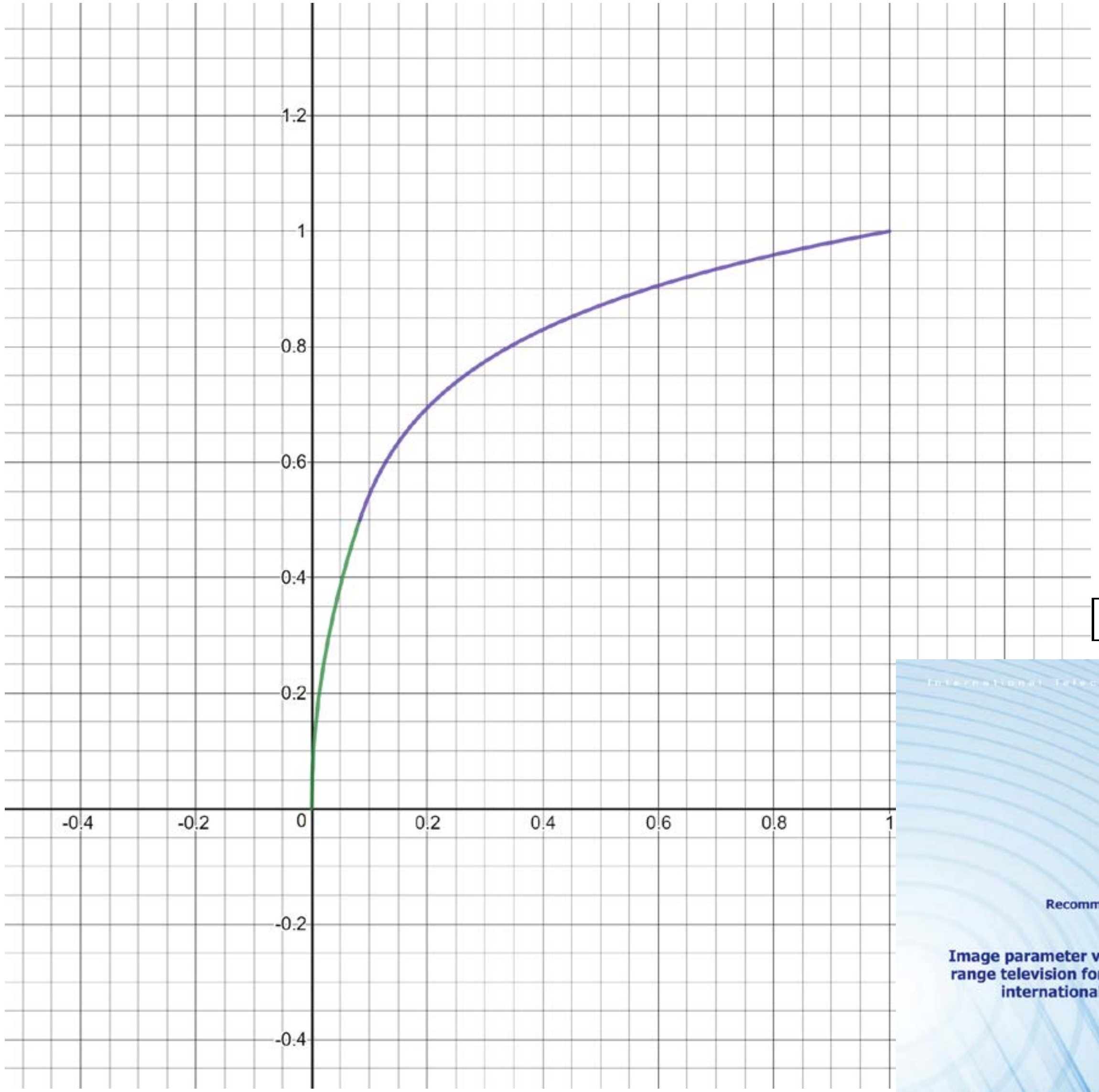
- Standard of HDR TV.
- Based on Dolby Vision curve.
- Encodes absolute signal.
- 10,000 nits maximum.
- “ITU-R Recommendation BT.2100 HDR10 EOTF” = ST2084 = PQ Curve



[BT2100]

Hybrid Log Gamma (OETF)

- Some compatibility to SDR TVs.
- Encodes relative signal.
- Dynamic range x10 of SDR.



[BT2100]

Inverse PQ Curve

- Games usually use OETF.
- PQ curve defines an EOTF.
- Inverse PQ Curve is more used in game graphics output.
- We will use “Inverse PQ Curve” as a OETF for HDR TV.

1. Introduction of This Course (5 min.)
2. Introduction of Color and HDR (25 min.)
    a. Mechanism of Color Sensitivity
    b. High Dynamic Range Display
    c. Mechanism of Image transfer
    d. Wide Color Gamut
3. Practical HDR Output Techniques (10 min.)
4. Asset Creation (15 min.)
5. Implementations and Results in Our Game (35 min.)
6. HDR Working Environment (10 min.)

What is Wide Color Gamut (WCG)?

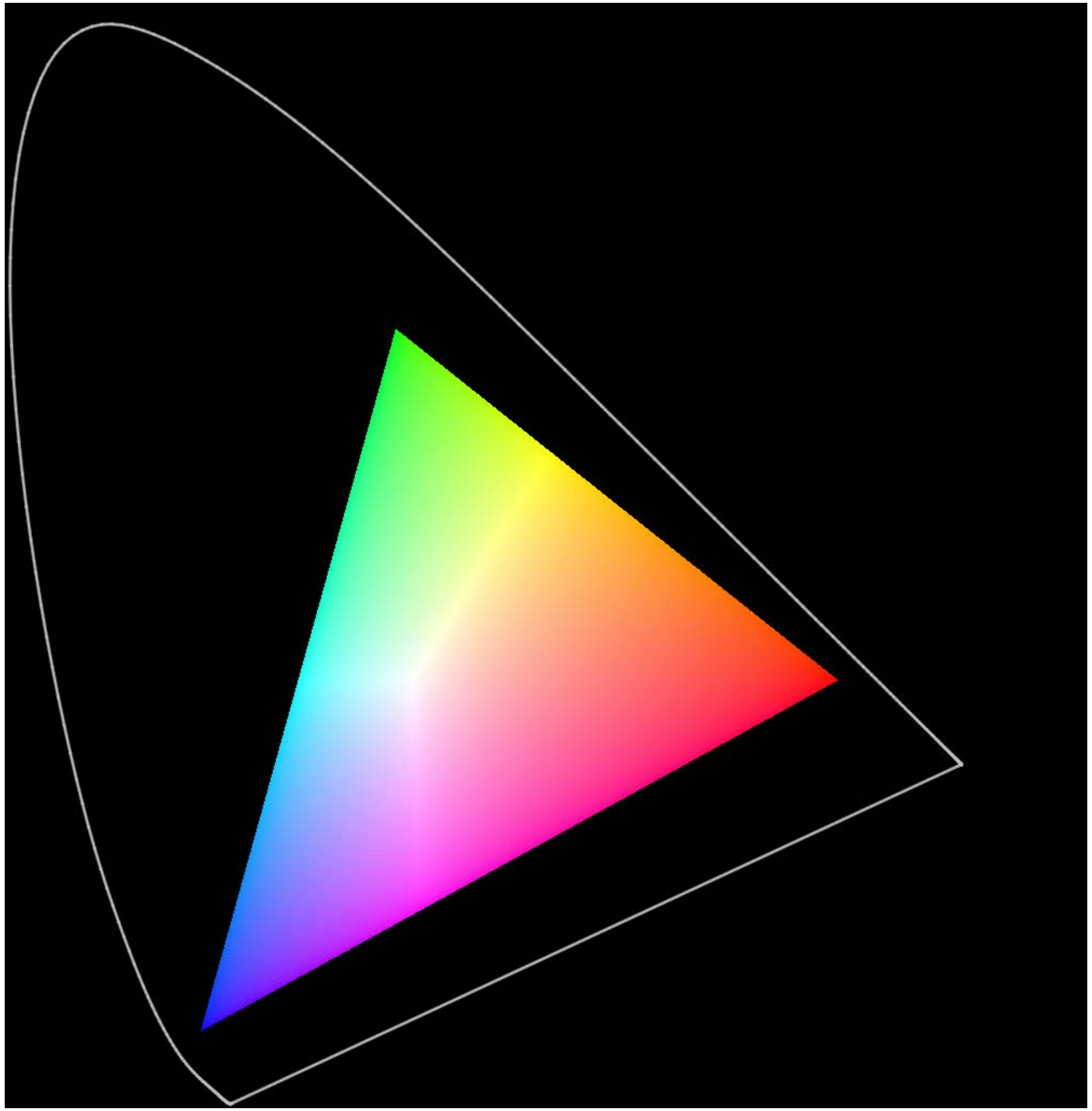
- Wider color gamut than conventional sRGB.

Why We Use Wide Color Gamut?

- Why not?
- The world is full of colors.
- sRGB is not sufficient.

sRGB

	x	y
White	0.3127	0.3290
Red	0.64	0.33
Green	0.30	0.60
Blue	0.15	0.06



Wide Color Gamut

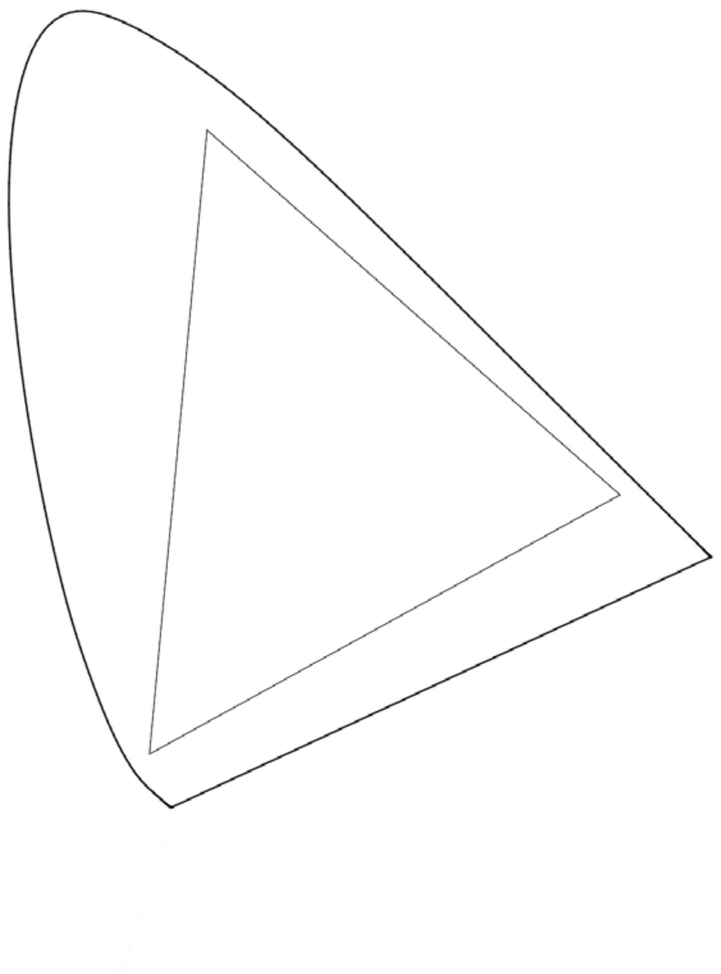
- There are so many gamut proposed.
- Here is a list of popular wide color gamut.

	Adobe	sRGB (reference)	**BT.2020**	**ACES2065-1**	**ACEScg**
xr	0.64	0.64	0.708	0.7347	0.713
yr	0.33	0.33	0.292	0.2653	0.293
xg	0.21	0.30	0.170	0	0.165
yg	0.71	0.60	0.797	1	0.83
xb	0.15	0.15	0.131	0.0001	0.128
yb	0.06	0.06	0.046	-0.077	0.044
White	D65	D65	D65	D60	D60

Adobe RGB

- Most professional monitors and digital cameras support this gamut.
- Only green is wider than sRGB.
- Good for first step.

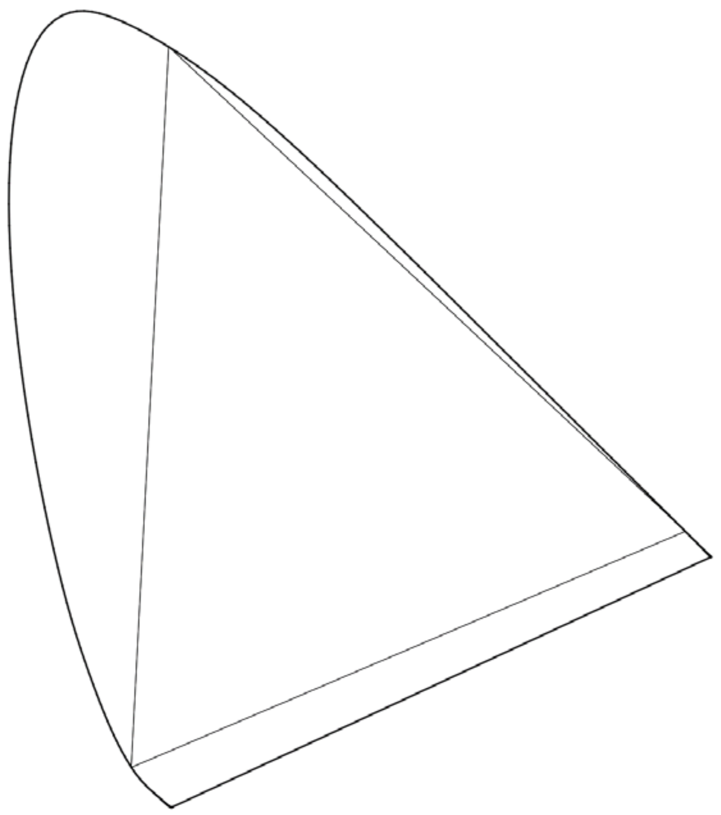
	x	y
White	0.3127	0.3290
Red	0.64	0.33
Green	0.21	0.71
Blue	0.15	0.06



BT.2020

- HDR TV standard.
- All primary colors are on locus.
- No TV output this gamut yet.

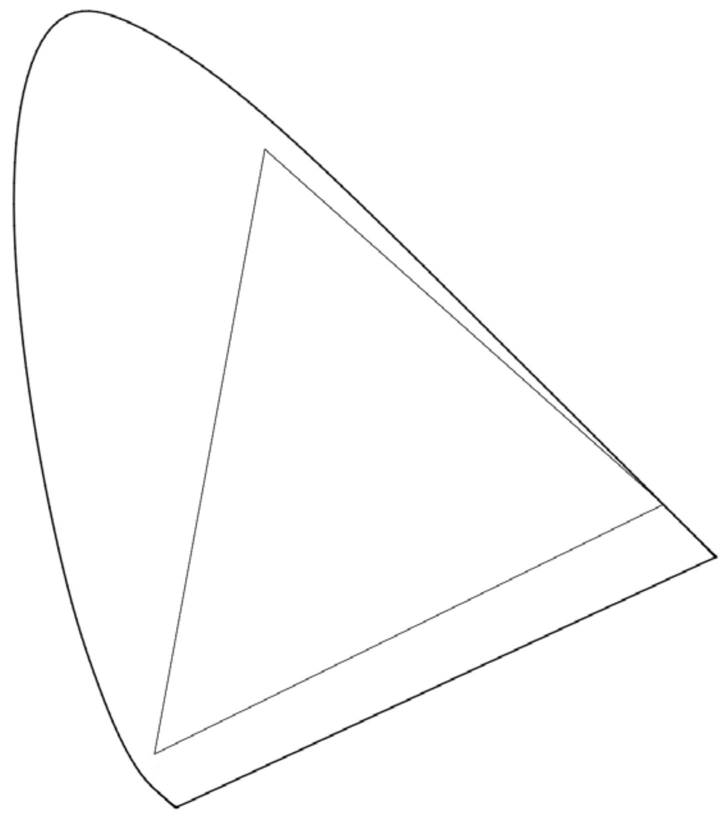
	x	y
White	0.3127	0.3290
Red	0.708	0.292
Green	0.170	0.797
Blue	0.131	0.046



DCI-P3

- iPhone supports this gamut.
- Some good monitor also support.
- Mostly used in Cinema grading.

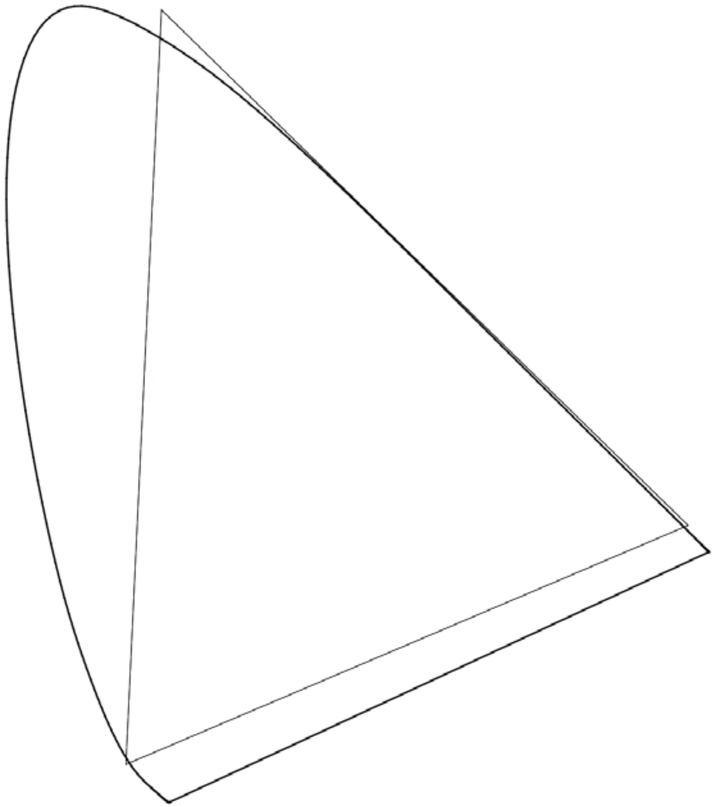
	x	y
White	0.3127 / 0.314	0.3290 / 0.351
Red	0.68	0.32
Green	0.265	0.69
Blue	0.15	0.06



ACEScg

- Academy Color Encoding System
- Slightly larger than BT.2020.
- Designed for rendering and compositing.

	x	y
White	0.32168	0.33767
Red	0.713	0.293
Green	0.165	0.830
Blue	0.128	0.044

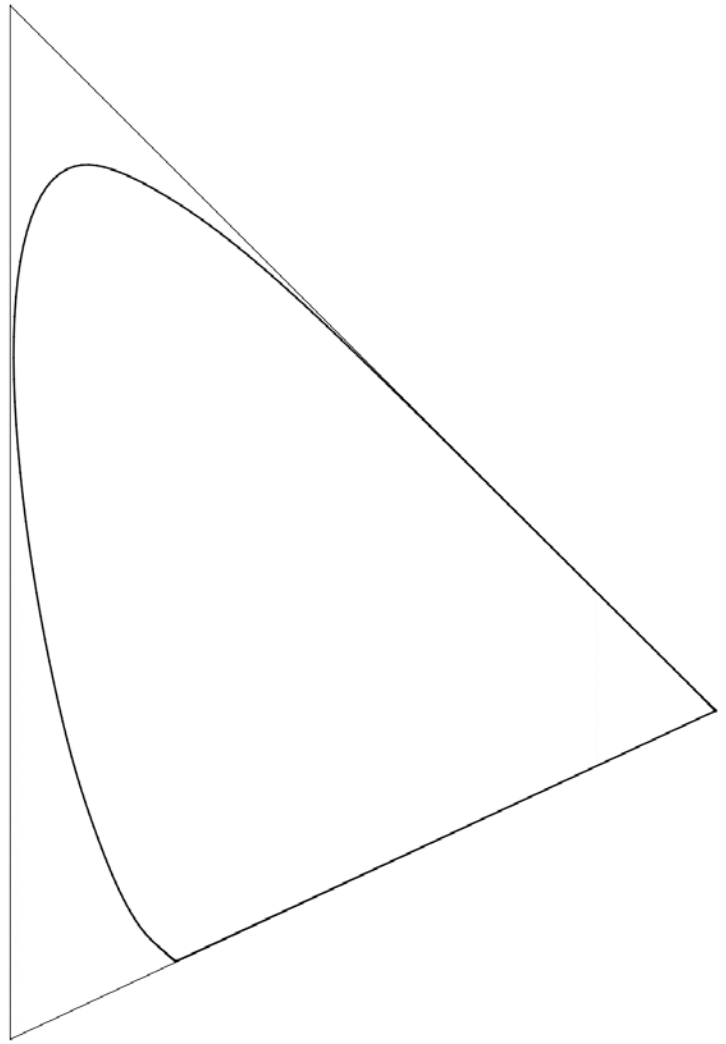


http://duikerresearch.com/2015/09/acescg-a-common-color-encoding-for-visual-effects-applications/

ACES2065-1

- Designed for storing image.
- Standardized by SMPTE as document ST2065-1.
- Covers full XYZ space.

	x	y
White	0.32168	0.33767
Red	0.7347	0.2653
Green	0.0000	1.0000
Blue	0.0001	-0.0770



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

Benefits of Wide Color Gamut

- Better representation of most artificial lights
  - Traffic signal, Neon
- Better quality after heavy post process
  - For example : color temperature
  - 2500K-13000K with no saturation!
- Better lighting accuracy
  - Lighting calculation in WCG is better than sRGB.
- Benefits still work even output device is sRGB.

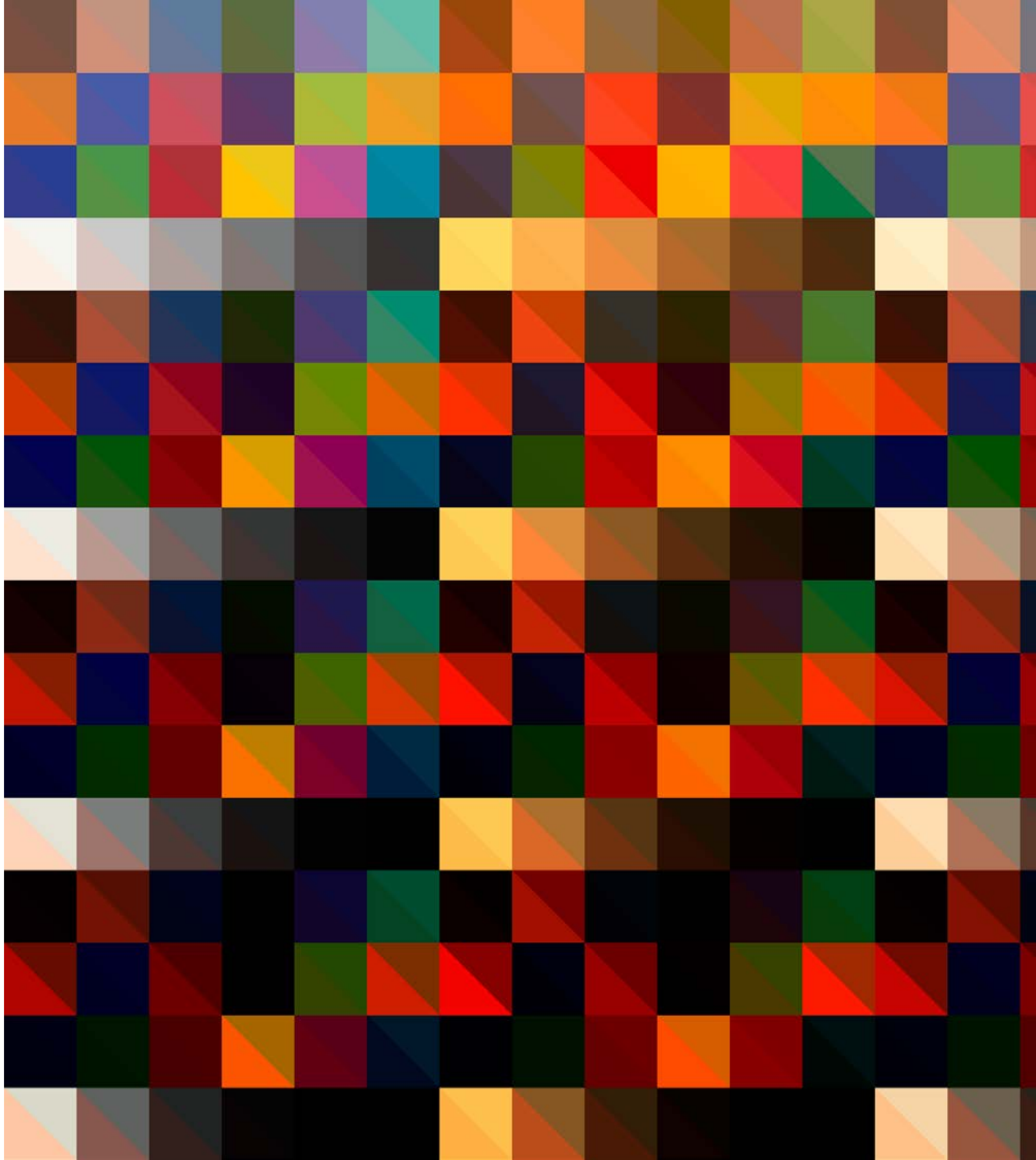
Better Lighting Accuracy

- ACEScg and BT.2020 is more accurate than sRGB.
  - Fewer clipped colors.
  - Evenly distributed hue.

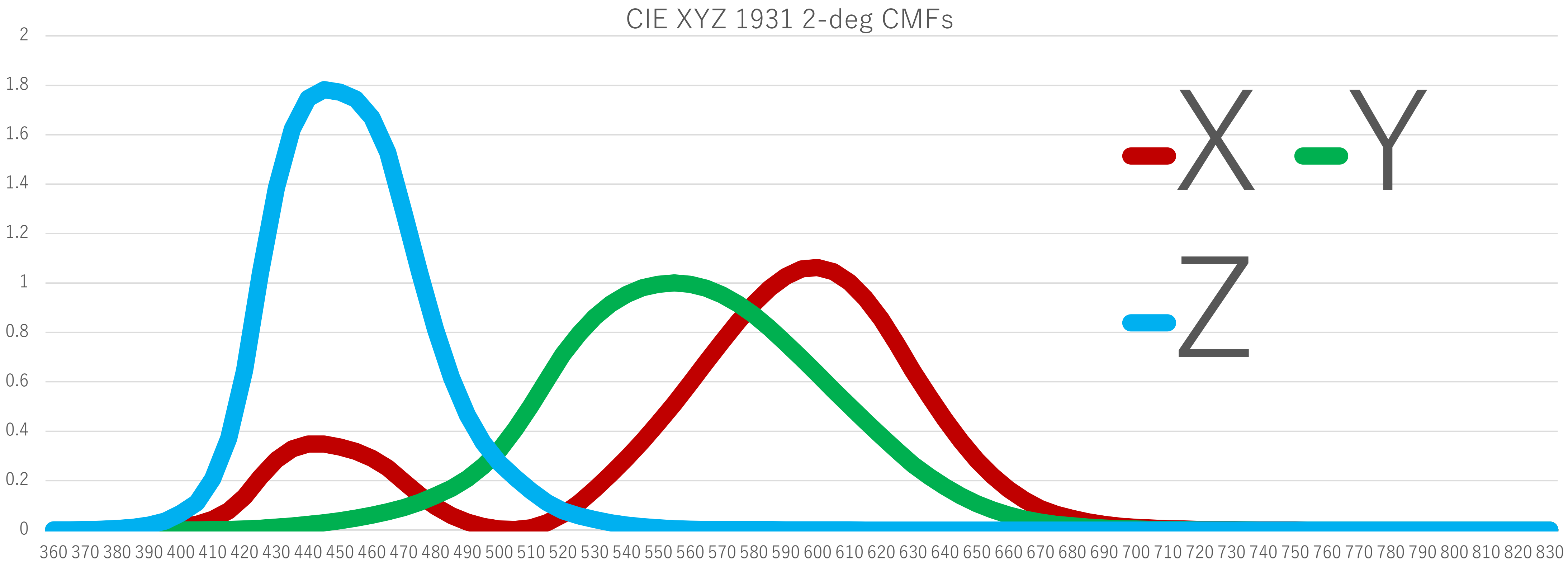
Let’s Do a Test

Tristimulus is RGB.
R*R, G*G, B*B.

Spectrum is f(λ).
Light(λ) * Reflectance(λ).

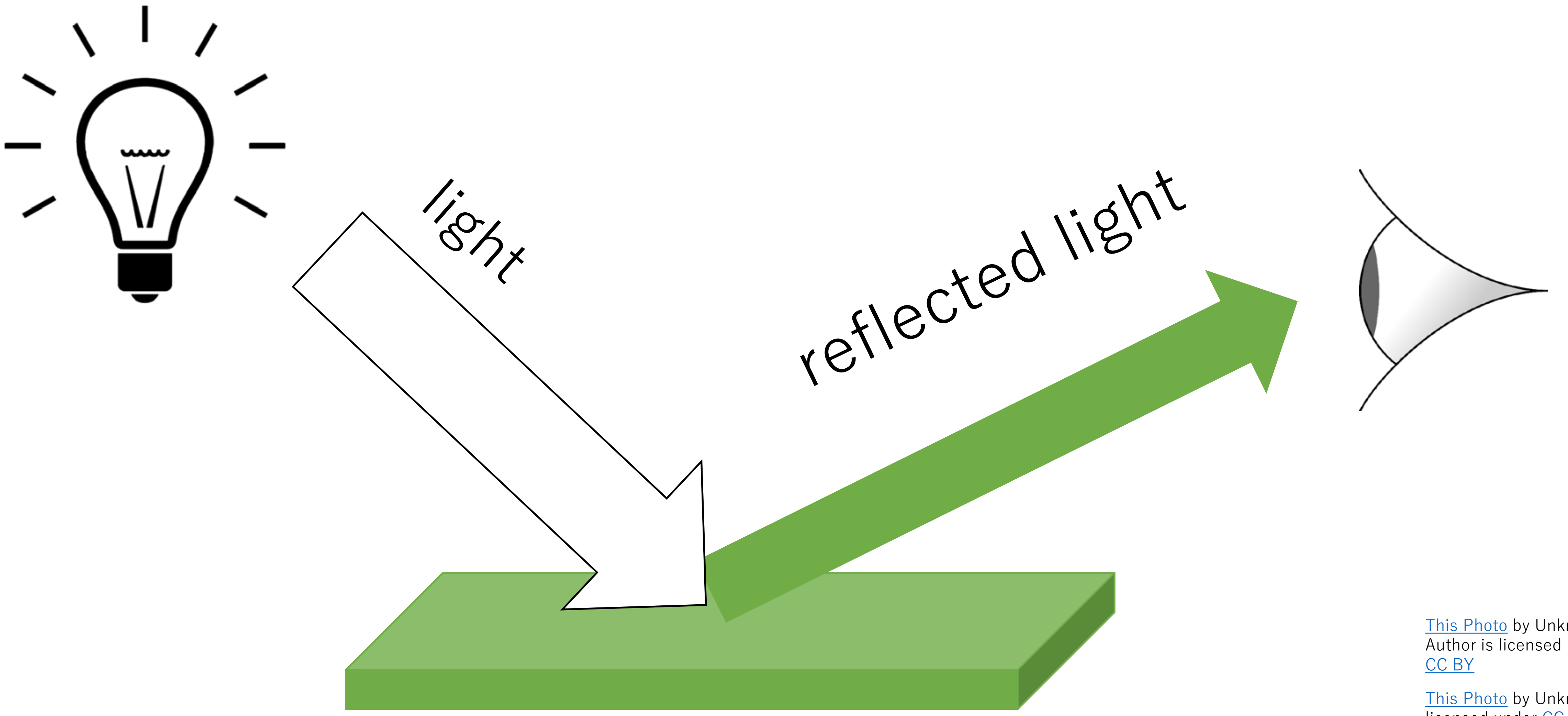


Color from spectrum with CMFs.

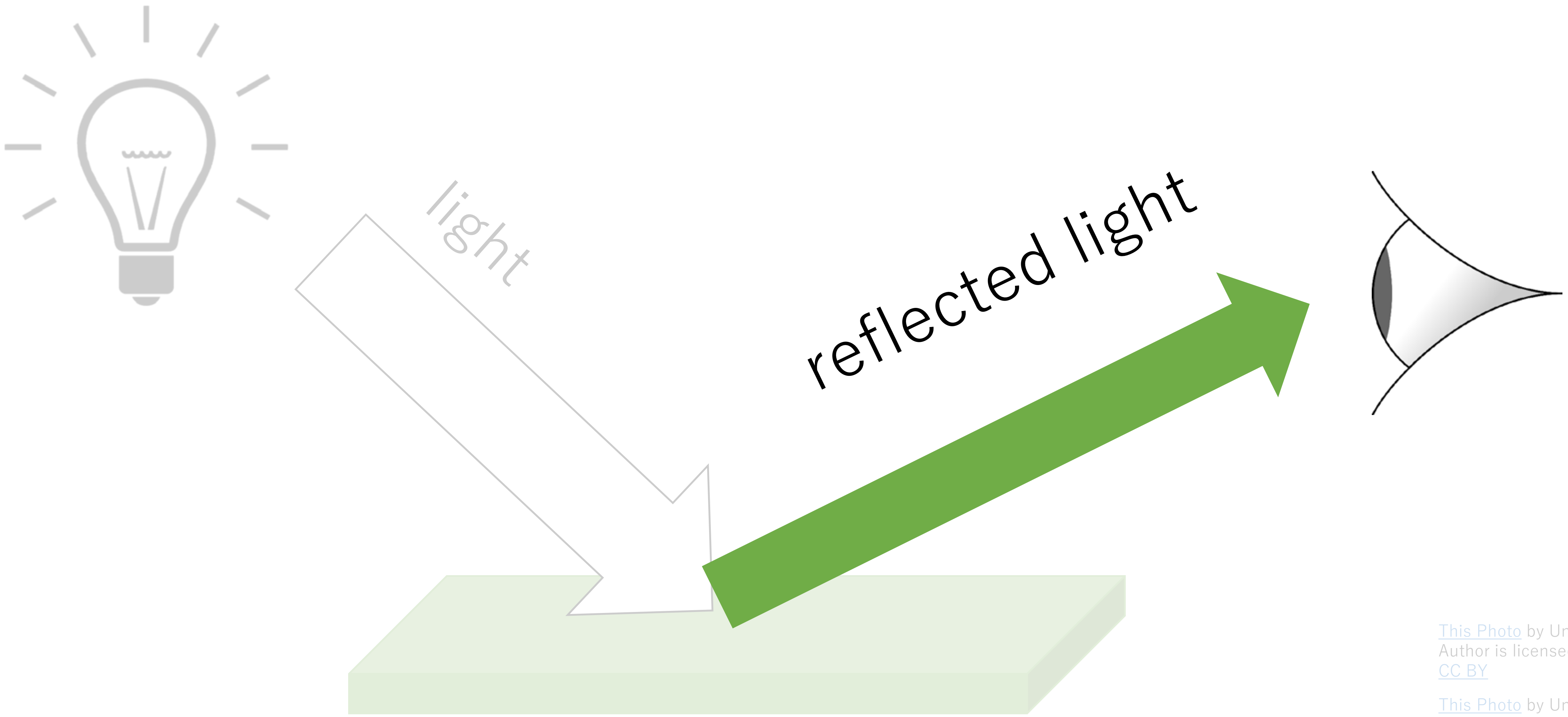


[CIE 1931]

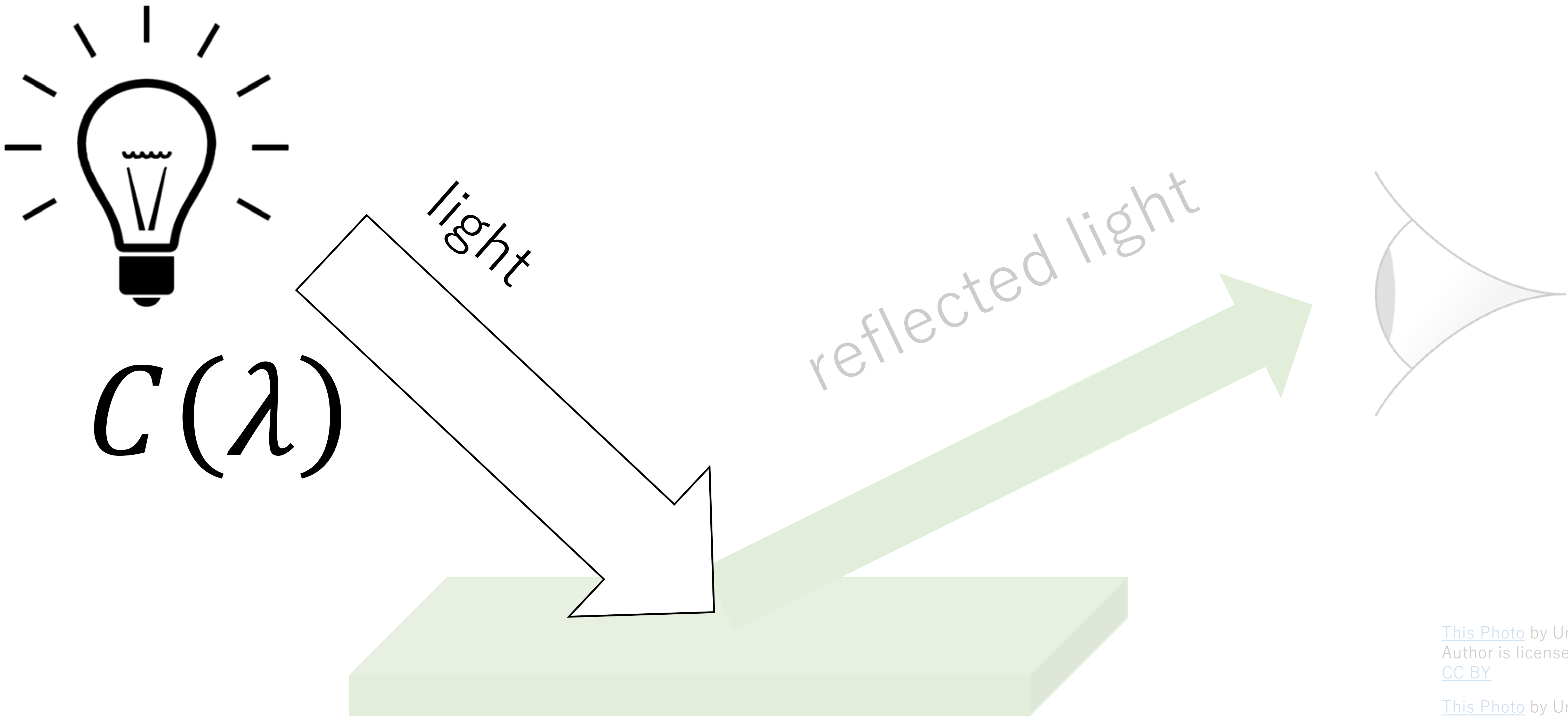
Simple Sample Setup



Calculate a Color of Reflected Light

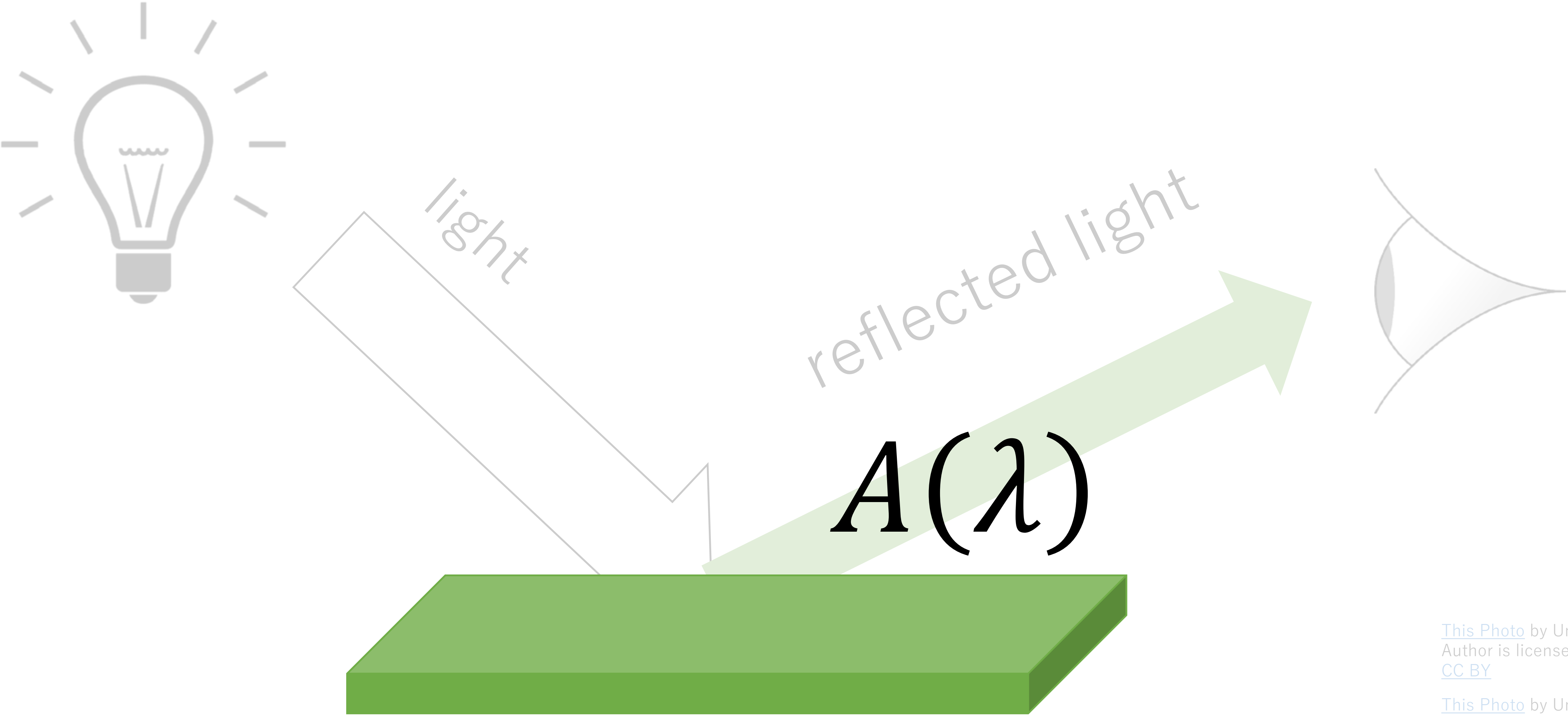


Illuminant Spectrum

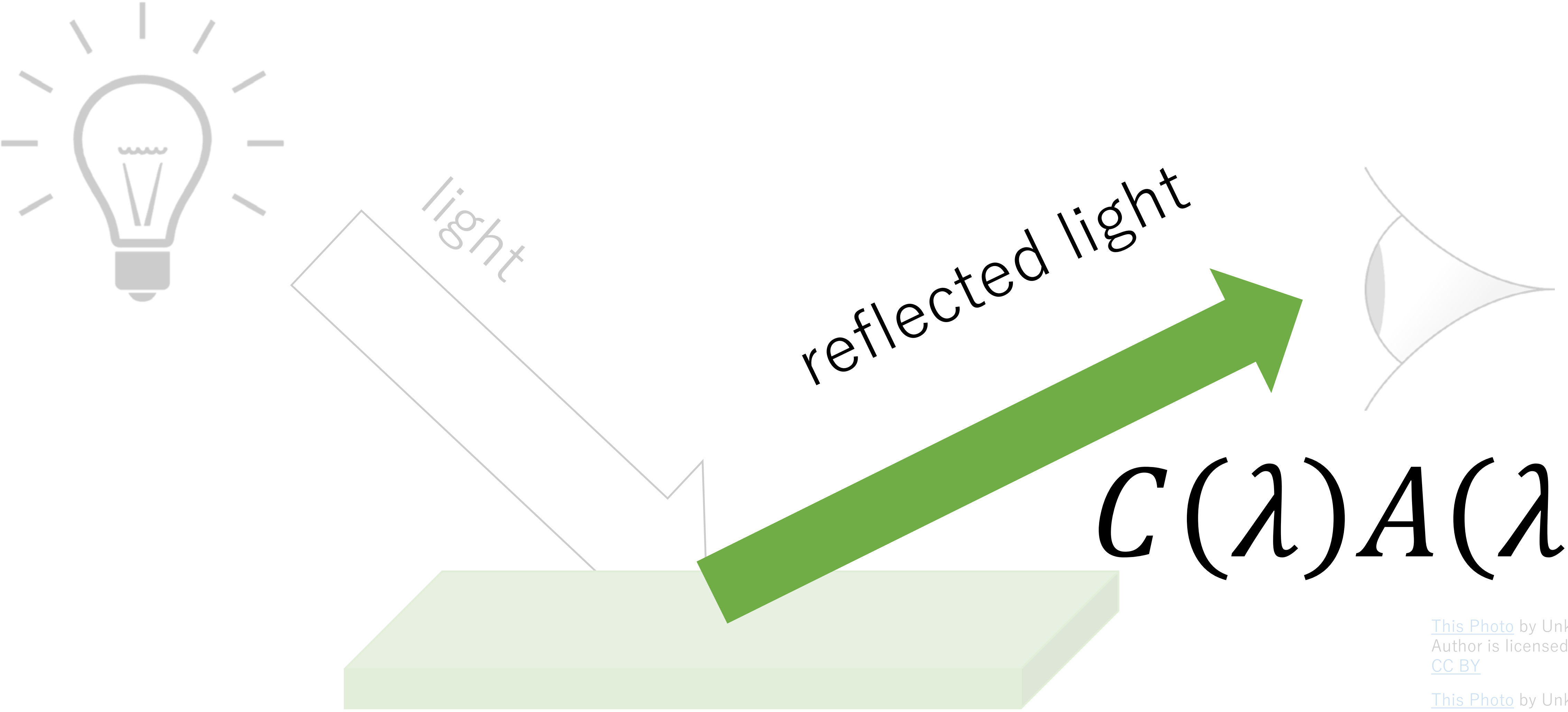


$c(\lambda)$

Reflectance Spectrum



Reflected Light



$$C(\lambda)A(\lambda)$$

The Color We See in Spectrum Domain

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} \int C(\lambda)A(\lambda)\bar{X}(\lambda)d\lambda \\ \int C(\lambda)A(\lambda)\bar{Y}(\lambda)d\lambda \\ \int C(\lambda)A(\lambda)\bar{Z}(\lambda)d\lambda \end{pmatrix}$$

The Color We See in Tristimulus

$$\begin{pmatrix} X_C \\ Y_C \\ Z_C \end{pmatrix} = \begin{pmatrix} \int C(\lambda)\bar{X}(\lambda)d\lambda \\ \int C(\lambda)\bar{Y}(\lambda)d\lambda \\ \int C(\lambda)\bar{Z}(\lambda)d\lambda \end{pmatrix}, \begin{pmatrix} X_A \\ Y_A \\ Z_A \end{pmatrix} = \begin{pmatrix} \int A(\lambda)\bar{X}(\lambda)d\lambda \\ \int A(\lambda)\bar{Y}(\lambda)d\lambda \\ \int A(\lambda)\bar{Z}(\lambda)d\lambda \end{pmatrix}$$

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} X_C * X_A \\ Y_C * Y_A \\ Z_C * Z_A \end{pmatrix}$$

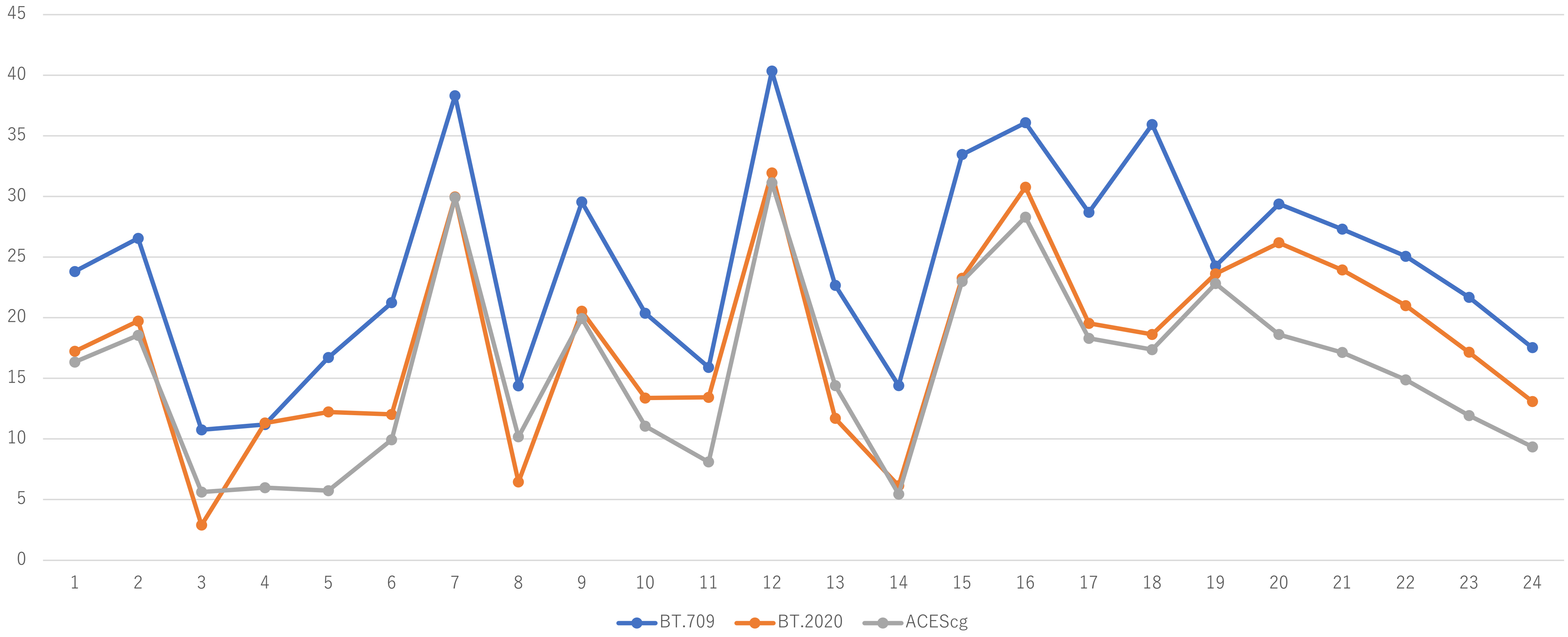
Result of BT.2020

Right top : Full spectrum
Left bottom : BT.2020

Result of sRGB

Right top : Full spectrum
Left bottom : sRGB

Color Differences



Average Error

sRGB	BT.2020	ACEScg
24.39502	17.75202	15.58134

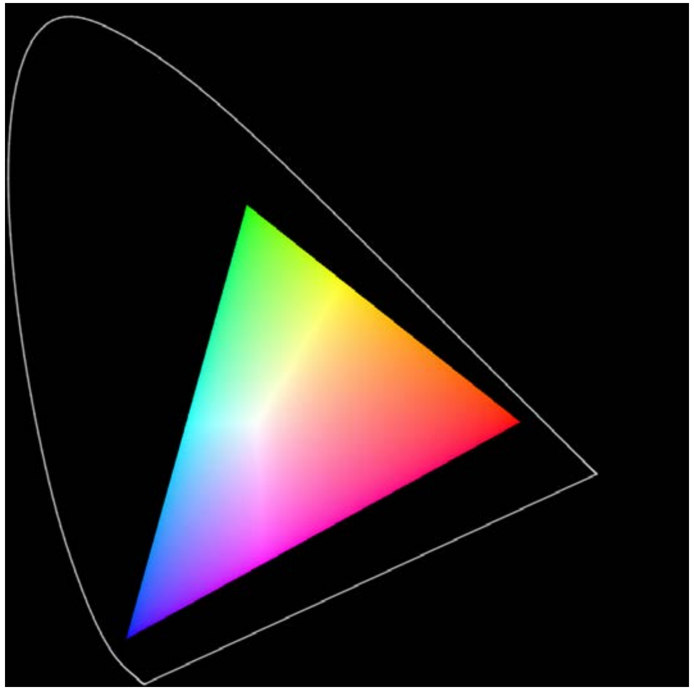
1. Introduction of This Course (5 min.)
2. Introduction of Color and HDR (25 min.)
3. **Practical HDR Output Techniques (10 min.)**
4. Asset Creation (15 min.)
5. Implementations and Results in Our Game (30 min.)
6. HDR Working Environment (10 min.)

HDR10

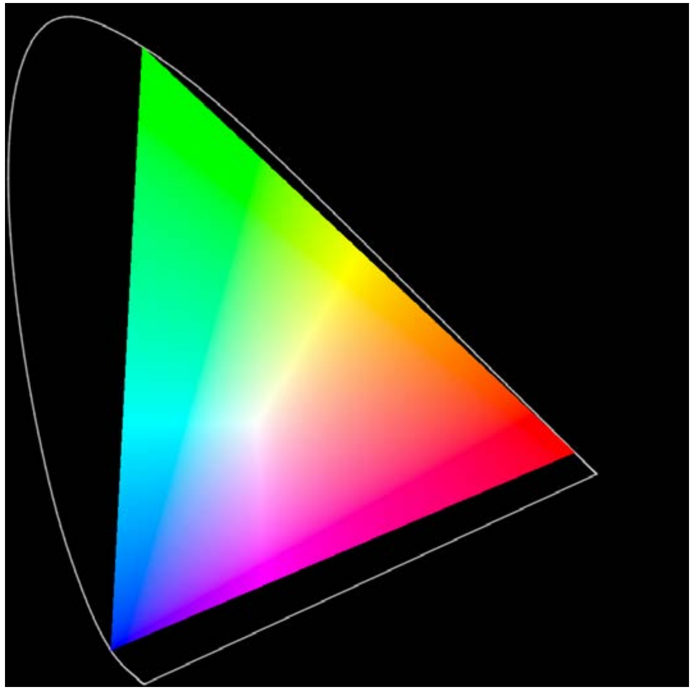
- We selected HDR10 to used in the game.
- HDR10 has metadata.

Gamut: Side by Side

- **SDR: sRGB**



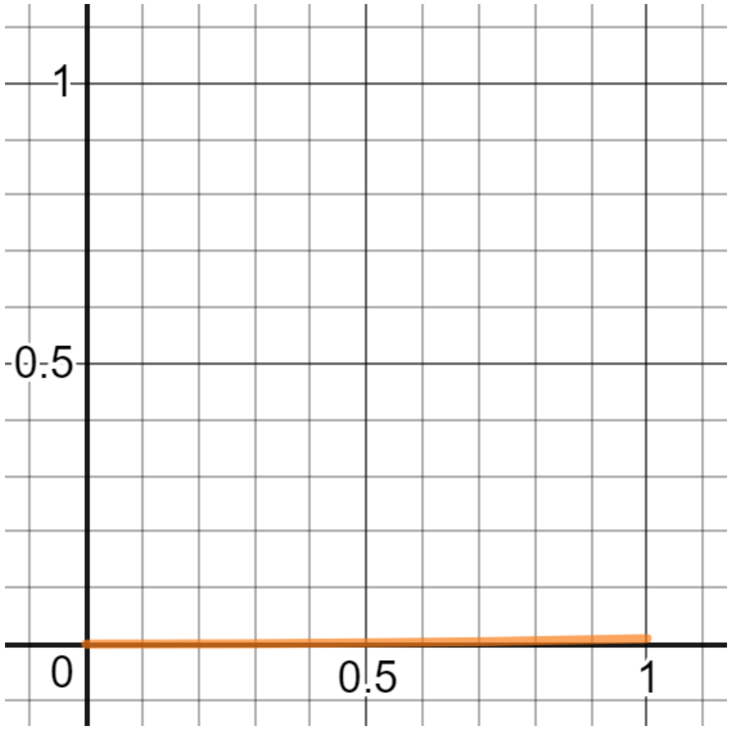
- **HDR: BT.2020**



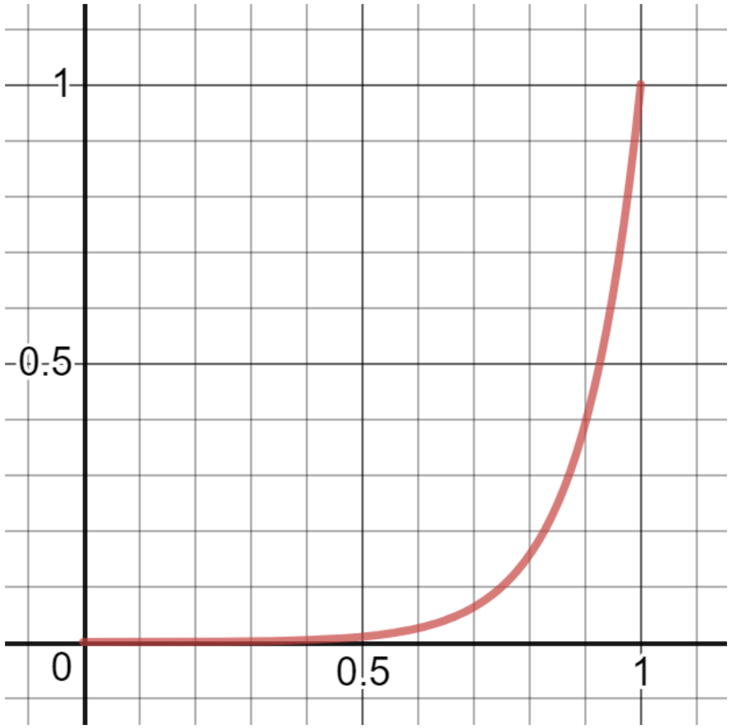
[BT2100][BT1886]

EOTF: Side by Side

- **SDR: sRGB gamma**



- **HDR: PQ Curve**



[BT2100][BT1886]

HDR10 Metadata

- **MaxFALL**
  - Maximum Frame Average Luminance Level.
- **MaxCLL**
  - Maximum Content Luminance Level.

MaxFALL

- MaxFALL of a picture frame can calculated like this:

$$\mathrm{m} = \frac{\sum_N \max\left(EOTF(R_i), EOTF(G_i), EOTF(B_i)\right)}{N}$$

- N: pixel count
- R_i, G_i, B_i: pixel value of each channel
- $EOTF(x)$: PQ Curve

MaxCLL

- MaxCLL of a picture frame can calculated like this:

$$\mathrm{m} = \max\left(\max(EOTF(R_i), EOTF(G_i), EOTF(B_i))\right)$$

- R_i, G_i, B_i: pixel value of each channel
- $EOTF(x)$: PQ Curve

1. Introduction of This Course (5 min.)
2. Introduction of Color and HDR (25 min.)
3. Practical HDR Output Techniques (10 min.)
    a. Single Source Multiple Output
    b. Tone Mapping
    c. Gamut Mapping
    d. Basic Tips Moving SDR to HDR
4. Asset Creation (15 min.)
5. Implementations and Results in Our Game (30 min.)
6. HDR Working Environment (10 min.)

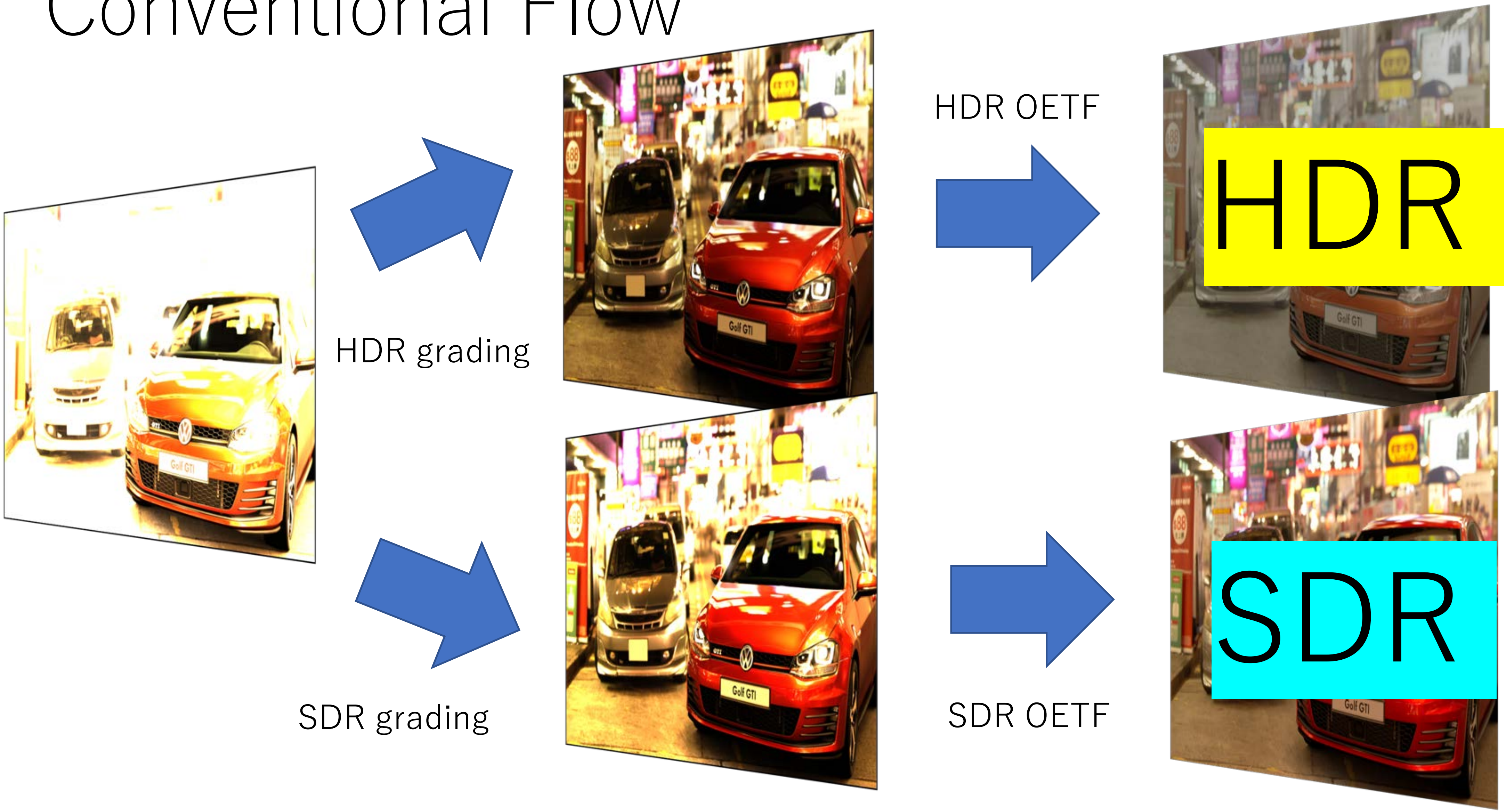
Different Gamut, Different OETF

- HDR OETF is not the same as the SDR OETF.
- HDR gamut is not the same as the SDR gamut.
- We have to apply different conversion functions.

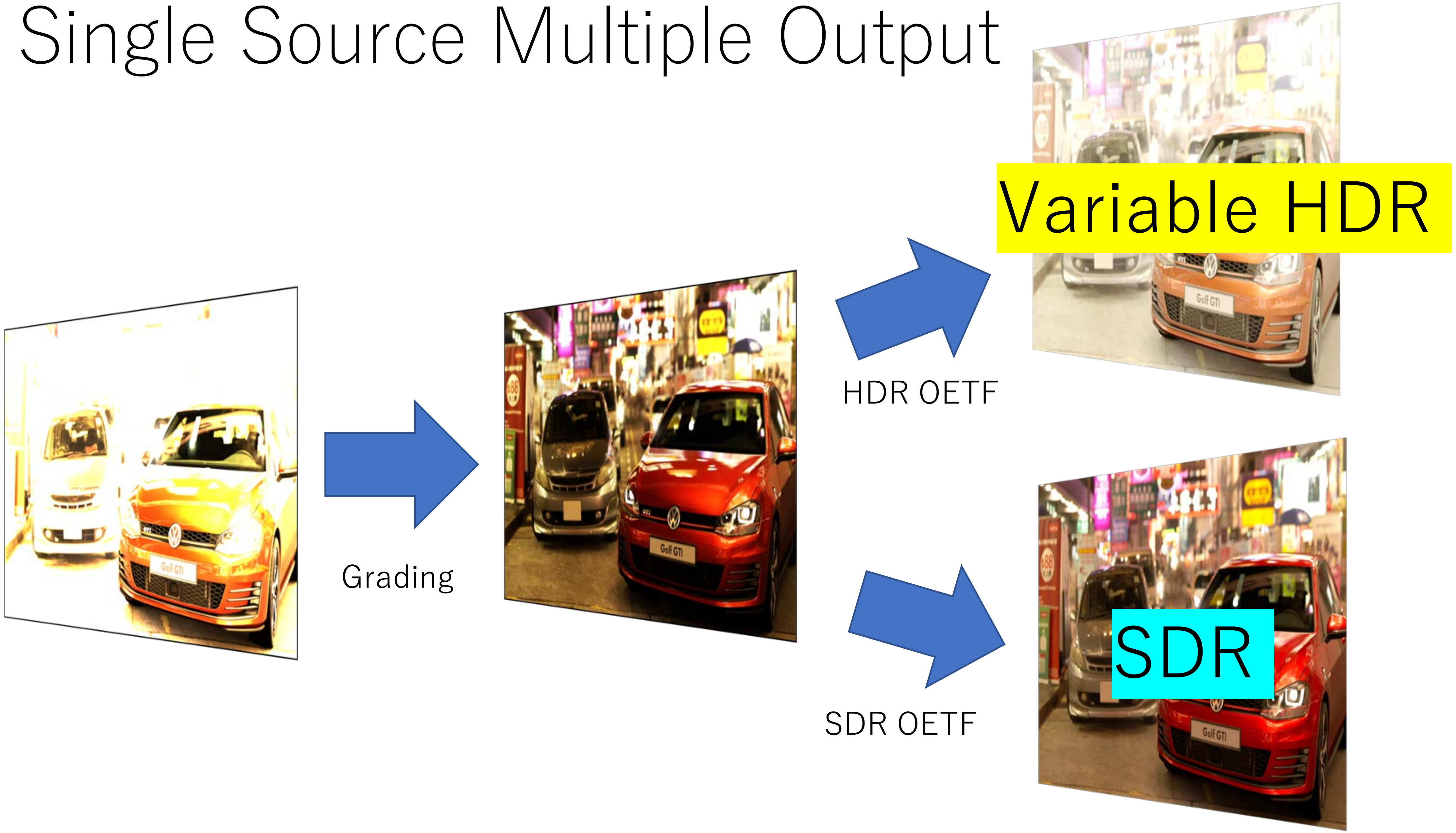
Single Source Multiple Output

- Game content is dynamic and unpredictable.
- Can not apply static method.
- Single source multiple target.

Conventional Flow



Single Source Multiple Output



Two Technique to Make SDR/HDR Compatibility

- **Variable tone mapping**
  - Make image smoother on HDR TVs with different brightness spec.
- **Gamut mapping**
  - Convert wide color image to fit in SDR TV.

1. Introduction of This Course (5 min.)
2. Introduction of Color and HDR (25 min.)
3. Practical HDR Output Techniques (10 min.)
    a. Single Source Multiple Output
    b. Tone Mapping
    c. Gamut Mapping
    d. Basic Tips Moving SDR to HDR
4. Asset Creation (15 min.)
5. Implementations and Results in Our Game (30 min.)
6. HDR Working Environment (10 min.)

Variable Tone Mapping

- All HDR TVs on the market have different brightness specifications.
  - Some TV can output 4,000 nits.
  - Some TV can output only 300 nits.

- So we need an adaptive mapping function for luminance.

Tone Mapping Variations

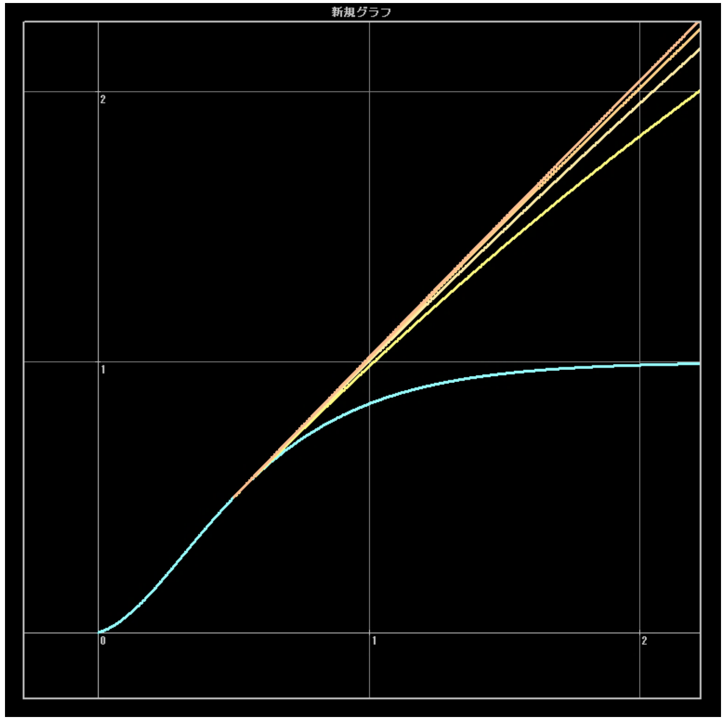
- Fixed tone mapping curve
  - ACES RRT+ODT
- Flexible, variable peak brightness tone mapping
  - Kawase's tone mapping[Kawase 2016]
  - Lottes's tone mapping[Lottes 2016]
- Both of them has limitations and not fit for us.
- So we developed our own tone mapping function.

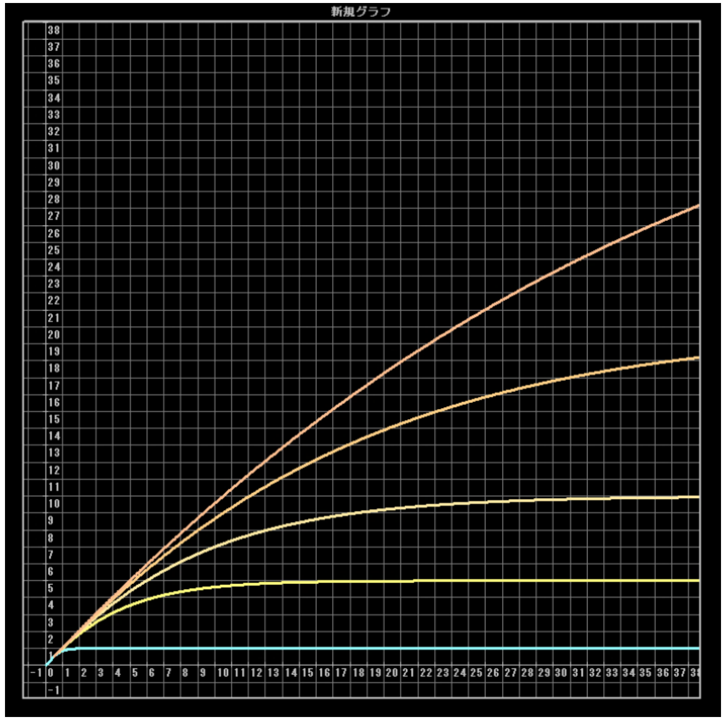
ACES RRT+ODT

- Based on Fujifilm's film characteristic.
  - RRT : Reference Rendering Transform
    - Transform scene linear color to film look color.
  - ODT : Output Device Transform
    - Transform scene linear color to device signal.
- Pros:
  - Battle proofed. Professional designed.
- Cons:
  - Fixed variations. ODT only support 1000/2000/4000nits display.
  - Filmic look baked in RRT.

Kawase Variable Tone Mapping

- First presented at CEDEC2016
  - **HDR Output Theory and Practice by Silicon Studio** [http://cedec.cesa.or.jp/2016/session/ENG/13532.html(Japanese)](http://cedec.cesa.or.jp/2016/session/ENG/13532.html) [Kawase 2016]
- From p74, he introduced a brief shape of tone mapping function with variable peak brightness.



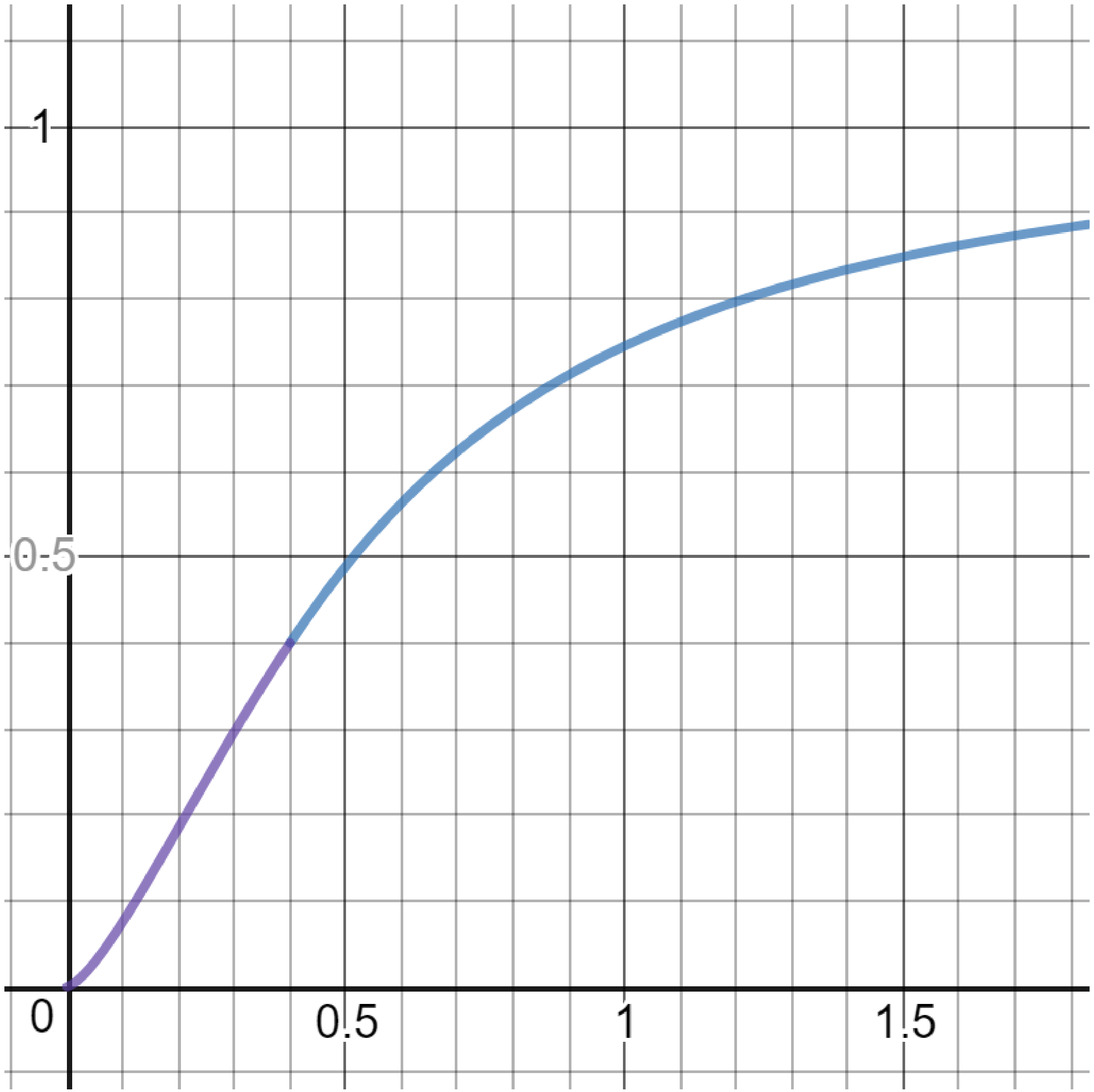


Kawase Tone Mapping

- Pros
  - Variable peak brightness!
  - Good idea to achieve HDR/SDR compatibility.
- Cons
  - No actual formulation.
  - No implementation information.

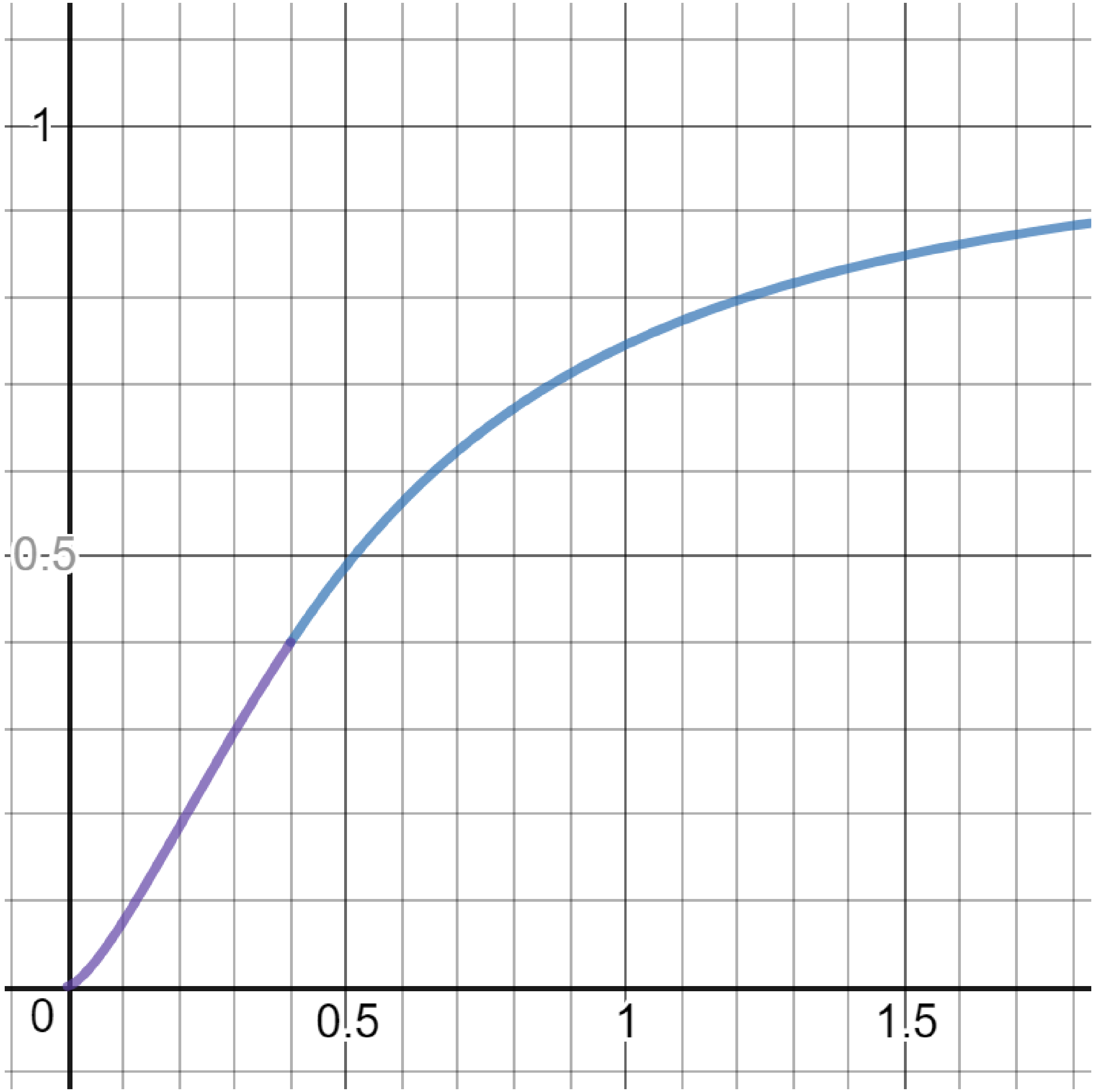
Lottes VDR Tone Mapping

- First presented at GDC2016
  - Advanced Techniques and Optimization of HDR VDR Color Pipelines https://gpuopen.com/gdc16-wrapup-presentations/ [Lottes 2016]
- He presented actual tone mapping function formula



Lottes Tone Mapping

- Pros
  - Formulations are open.
  - Variable peak brightness!
- Cons
  - Little bit difficult to adjust.
  - No linear part.



GT Tone Mapping

- We designed our own tone mapping function.

- Variable peak brightness!
- Linear mid makes picture photorealistic.
- Everything adjustable and very easy.

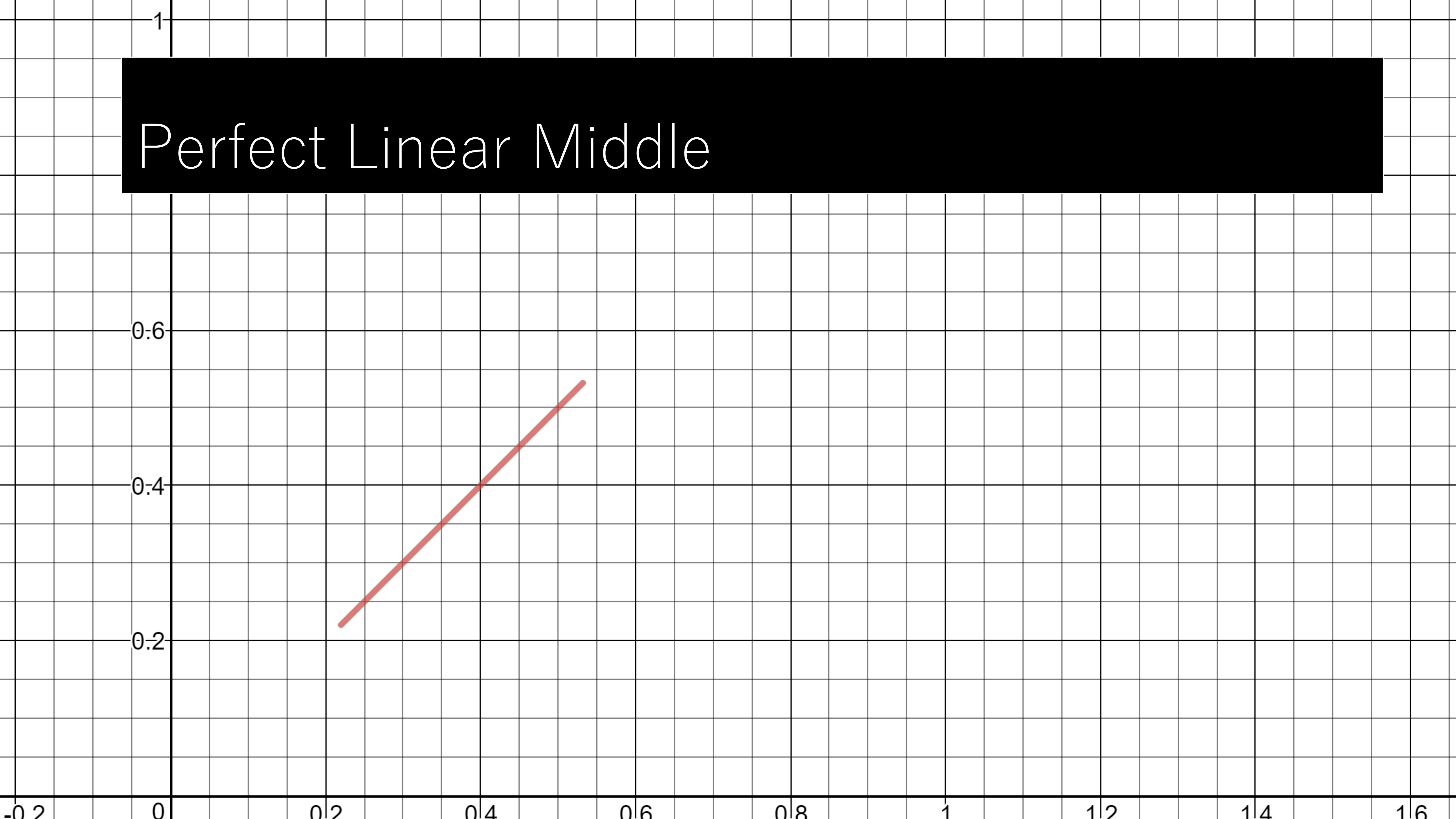
GT Tone Mapping Design Philosophy

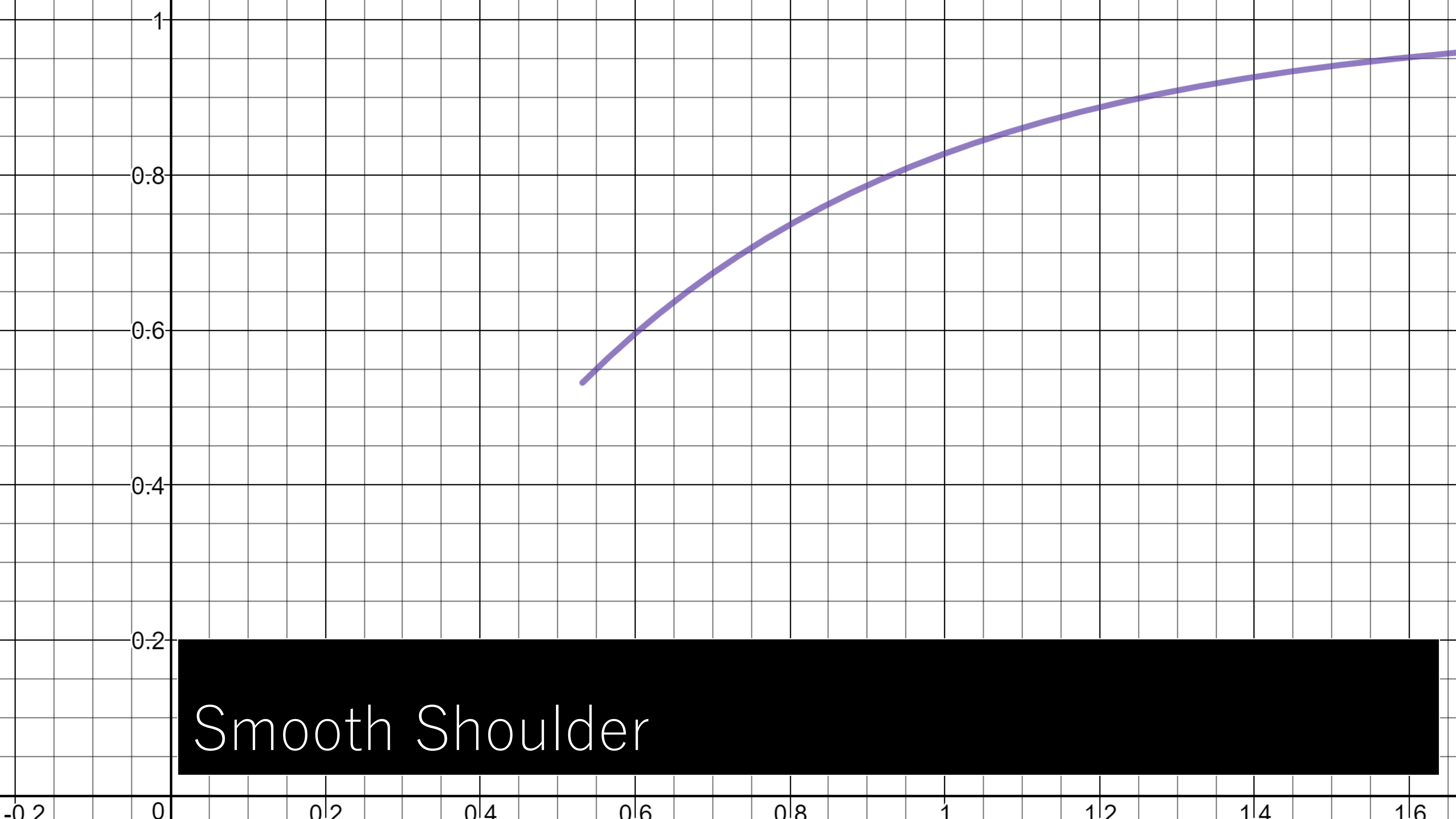
- Important three points we learnt from our artist:
    - Linear section in middle
    - Contrast adjustable toe
    - Smooth shoulder

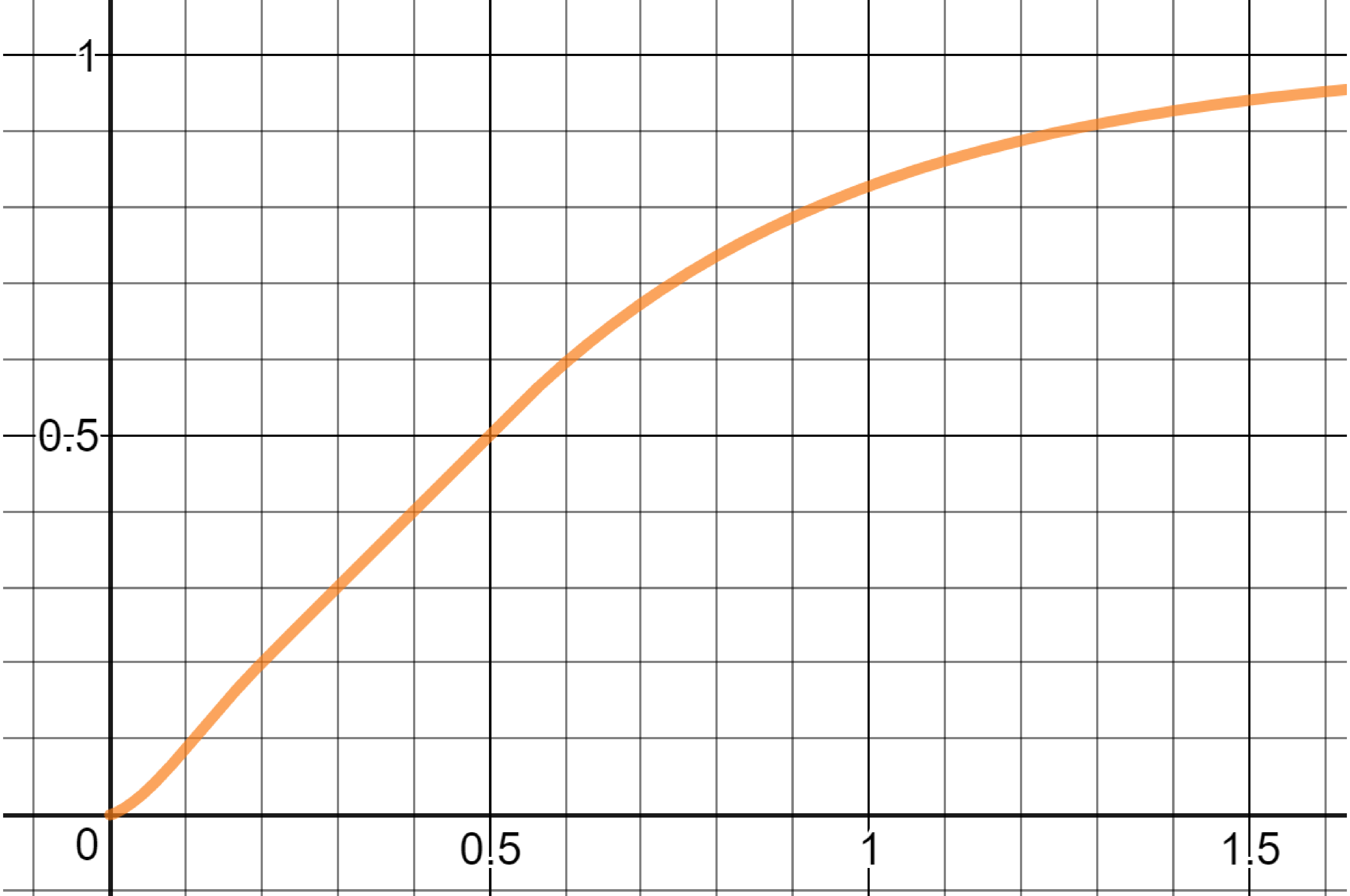
- GT Tone mapping smoothly connects these three sections.

Contrast Toe

Perfect Linear Middle

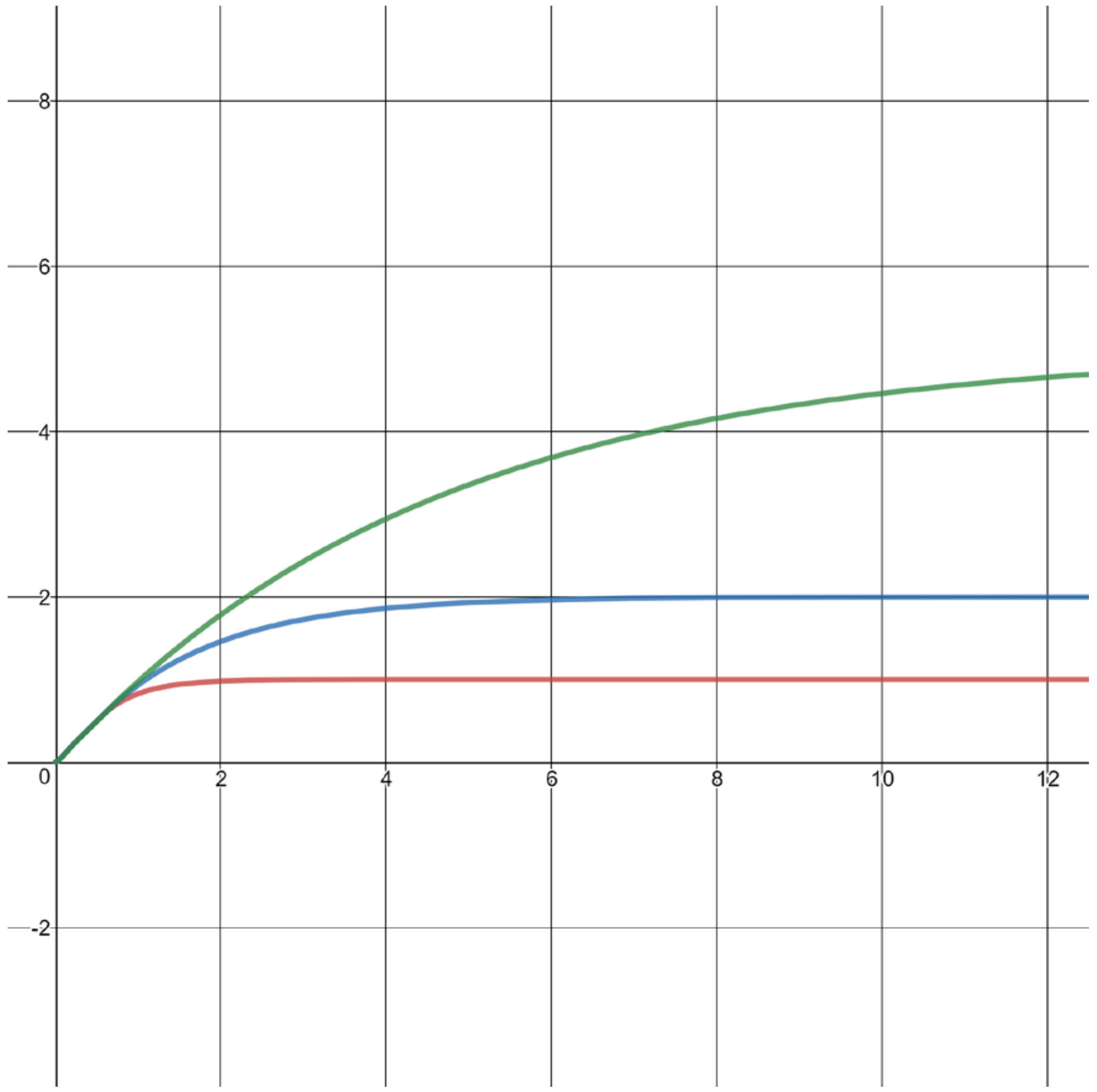



1
0.8
0.6
0.4
0.2
-0.2
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
Smooth Shoulder


1
0.5
0
0.5
1
1.5

GT Tone Mapping

- First presented at CEDEC2017 “HDR, Theory and Practice” [Uchimura 2017](Japanese)
  - https://www.desmos.com/calculator/mbkwnuihbd
  - https://www.slideshare.net/nikuque/hdr-theory-and-practicce-jp



Inverse Tone Mapping

- We used smoothstep function to connect those parts.
- Unfortunately, it makes inversion harder.

- In fact, the necessity of inversion function was almost missed at that time.

1. Introduction of This Course (5 min.)
2. Introduction of Color and HDR (25 min.)
3. Practical HDR Output Techniques (10 min.)
    a. Single Source Multiple Output
    b. Tone Mapping
    c. Gamut Mapping
    d. Basic Tips Moving SDR to HDR
4. Asset Creation (15 min.)
5. Implementations and Results in Our Game (30 min.)
6. HDR Working Environment (10 min.)

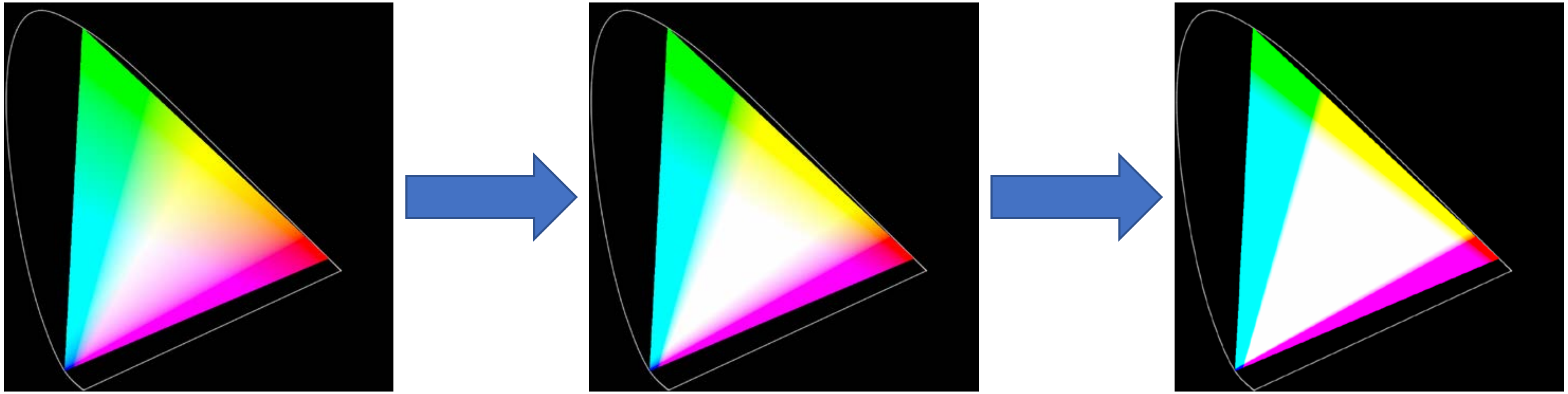
Gamut Mapping

- Simple gamut conversion makes noticeable artifact.
- Special care is required.

Why This Happens

- Simple conversion is okay at 100 nits.
- Brighter color will be clipped and saturated.

100 nits　　200 nits　　500 nits



Gamut Mapping in Games

- Frostbite Method [Fry 2017]
- GTSport Method(Ours) [Uchimura 2017]

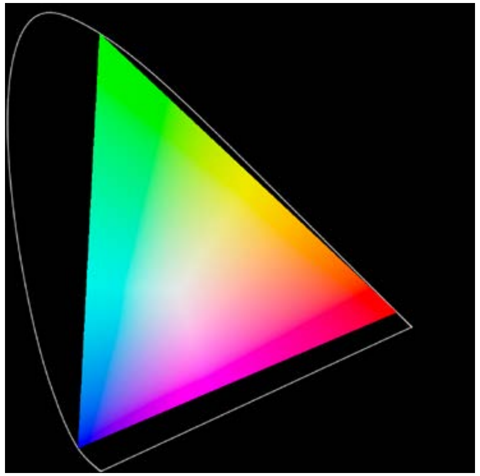
Frostbite Method

- HDR COLOR GRADING AND DISPLAY IN FROSTBITE[Fry 2017]
  - https://www.ea.com/frostbite/news/high-dynamic-range-color-grading-and-display-in-frostbite
- They used ICtCp [Dolby 2016] color space to split chromacity and lightness
- Lightness is tone mapped separately.
- Chroma desaturates with shoulder.

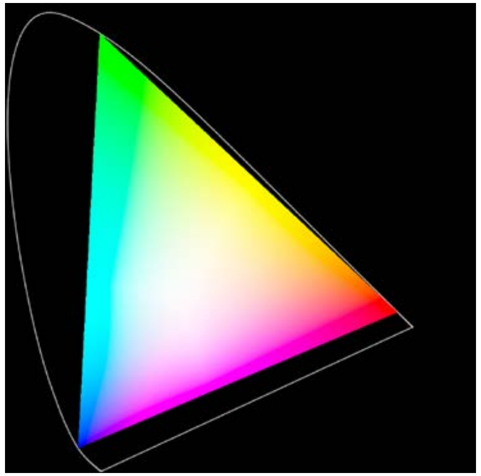
GTSport Method

- We apply GT Tone Mapping function to each RGB channel, achieving gamut mapping effect and tone mapping simulataneously.

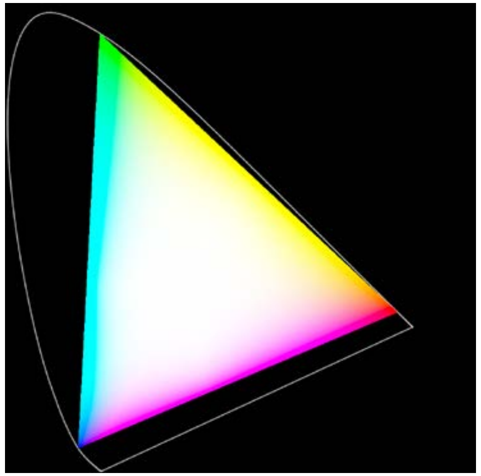
100 nits



250 nits



500 nits



Hue Shifting

- When applying tone map function for RGB channel, hue shifts at brighter area.

We Accepted

Plausible enough for us

Real Photo

1. Introduction of This Course (5 min.)
2. Introduction of Color and HDR (25 min.)
3. Practical HDR Output Techniques (10 min.)
    a. Single Source Multiple Output
    b. Tone Mapping
    c. Gamut Mapping
    d. Basic Tips Moving SDR to HDR
4. Asset Creation (15 min.)
5. Implementations and Results in Our Game (30 min.)
6. HDR Working Environment (10 min.)

First, know good HDR image.

- You will need to know what is “good” in HDR.

- You need a High-end HDR TV.
- Open Netflix. Let’s take “Chef’s Table: France” as an example.
  - It has good HDR pictures in it
  - The reason is it has very bright specular, and it is not so much graded.

Second, just output it.

- Changing OETF and applying BT.709 to BT.2020 matrix is good point to start.
- Check them on HDR TV.

Expand your asset

- Inverse tone mapping for shininess.
- Chroma saturation for wide gamut.

- Not best answer but still a good point to start with.

1. Introduction of This Course (5 min.)
2. Introduction of Color and HDR (25 min.)
3. Practical HDR Output Techniques (10 min.)
4. Asset Creation (15 min.)
5. Implementations and Results in Our Game (30 min.)
6. HDR Working Environment (10 min.)

1. Introduction of This Course (5 min.)
2. Introduction of Color and HDR (25 min.)
3. Practical HDR Output Techniques (10 min.)
4. Asset Creation (15 min.)
    a. Capturing HDR
    b. Capturing Wide Color
    c. Handling HDR Images and Videos
5. Implementations and Results in Our Game (30 min.)
6. HDR Working Environment (10 min.)

HDR: Image Bracketing

- Image bracketing is a popular technique for capturing HDR images.
- Shoot multiple exposures at once.
- Estimate actual luminance from image brackets.

Debevec Method

- From each pixel on image $P_i(x)$ we estimate actual irradiance $I(x)$ using Debevec's weighted sum method [Debevec and Malik 1997]

$$I(x) = \sum_{i=1} \frac{P_i(x) W(P_i(x))}{W(P_i(x))}$$

$$W(z) = \begin{cases} z - Z_{min} & if\ z \leq \frac{1}{2}(Z_{min} + Z_{max}) \\ Z_{max} - z & if\ z > \frac{1}{2}(Z_{min} + Z_{max}) \end{cases}$$

Problem and Solution

- Naïve implementation of Debevec method is not good at dark picture.
- Dark picture contains a lot of noise.
- Weighted sum boosts the noise.
- To solve this, we created our own weighting function.
- Also implemented wavelet denoiser.

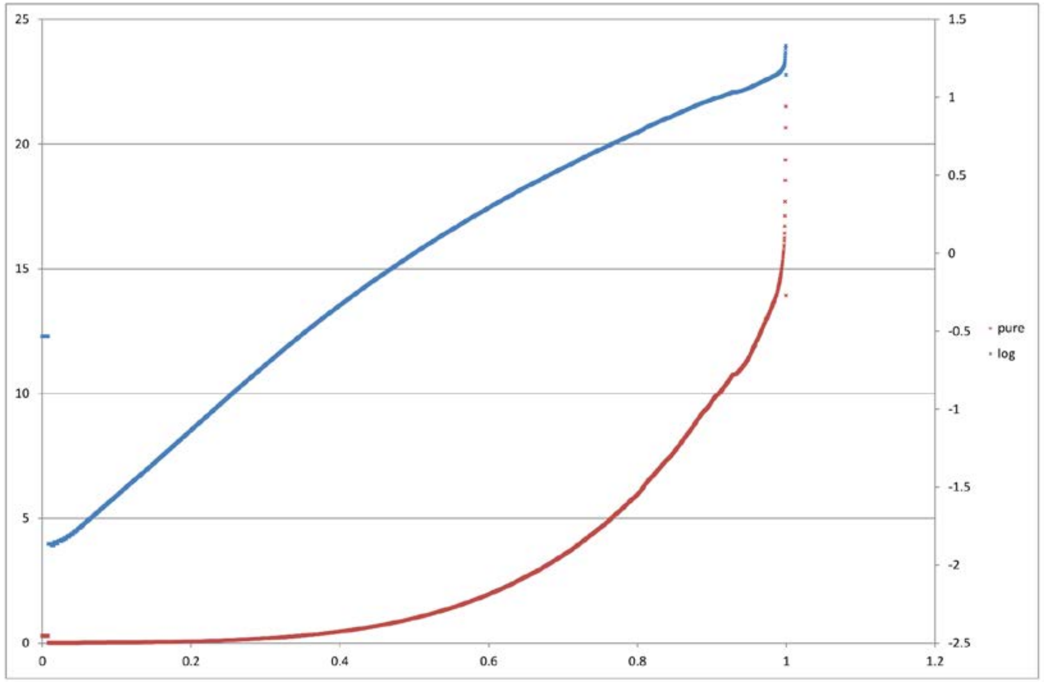
Importance of Capture Device Calibration

- To used in game, all image assets must be neutral as possible.
- No “look” or “develop” is required.

- A the calibration is very important.

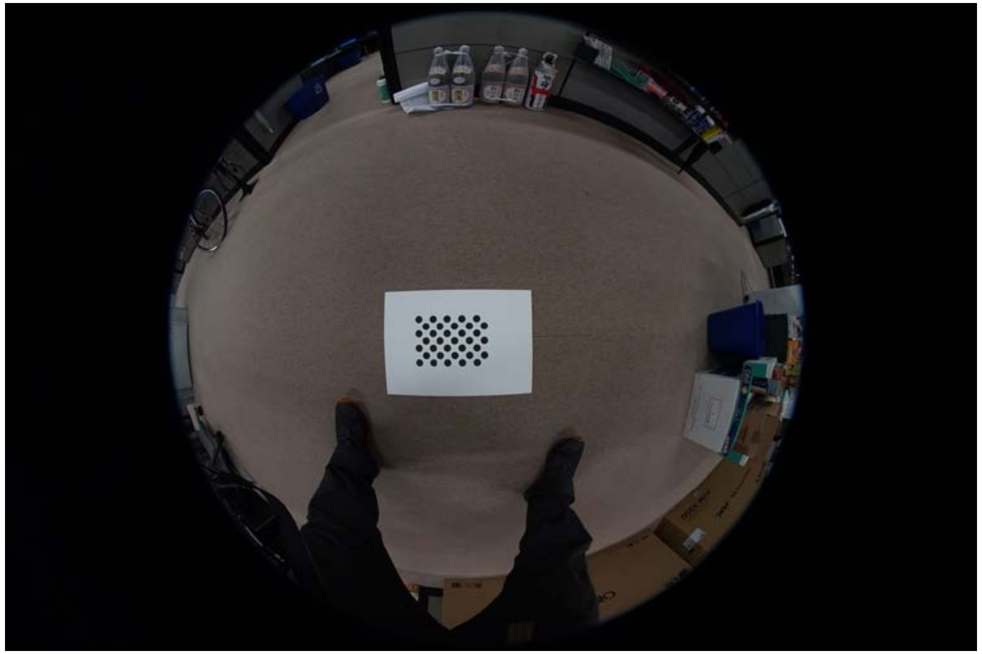
Sensor Calibration

- We used Mitsunaga method to calibrate image sensor linearity. [Mitsunaga and Nayar 1999]



Lens Calibration

- We used OpenCV to calibrate lens distortions. [OpenCV 2018]



1. Introduction of This Course (5 min.)
2. Introduction of Color and HDR (25 min.)
3. Practical HDR Output Techniques (10 min.)
4. Asset Creation (15 min.)
    a. Capturing HDR
    b. Capturing Wide Color
    c. Handling HDR Images and Videos
5. Implementations and Results in Our Game (30 min.)
6. HDR Working Environment (10 min.)

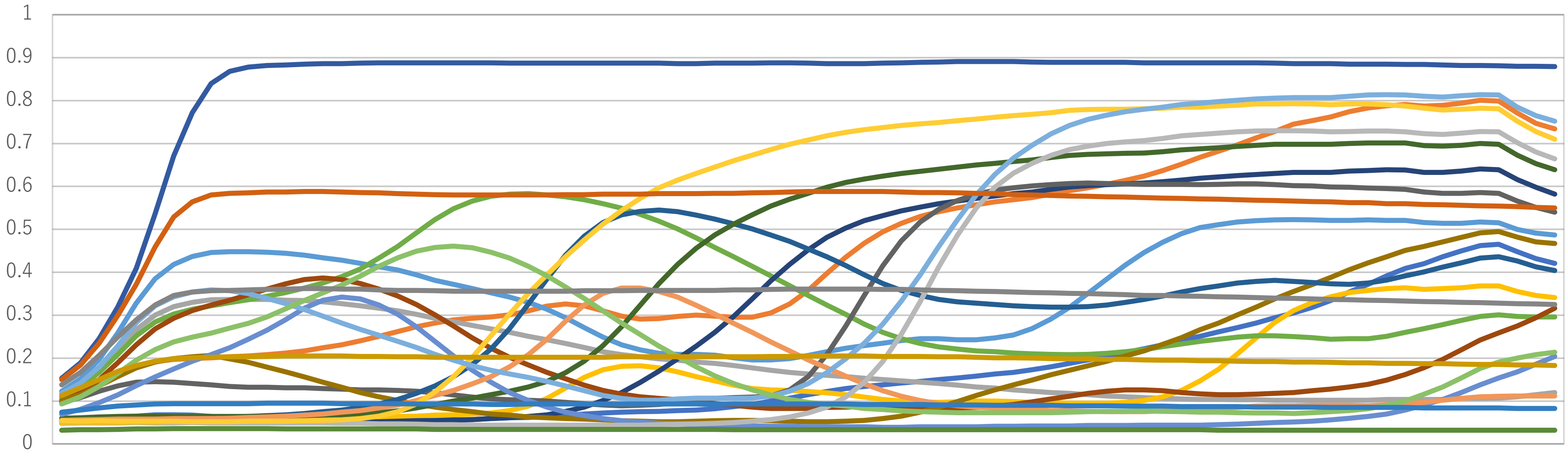
Color Calibration

- X-Rite Color Checker. [Xrite 2018]
- SEKONIC C-700 Spectrometer.


x·rite
colorchecker

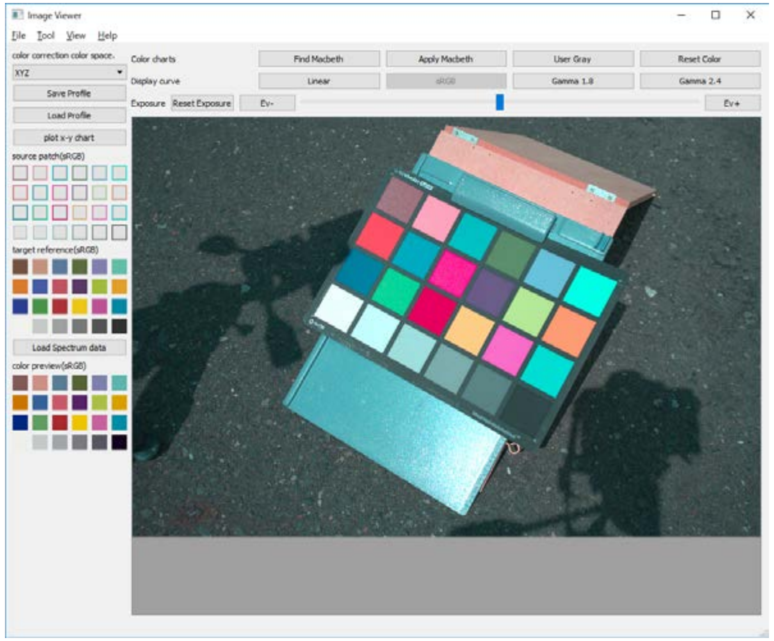
Color Checker Reflectance

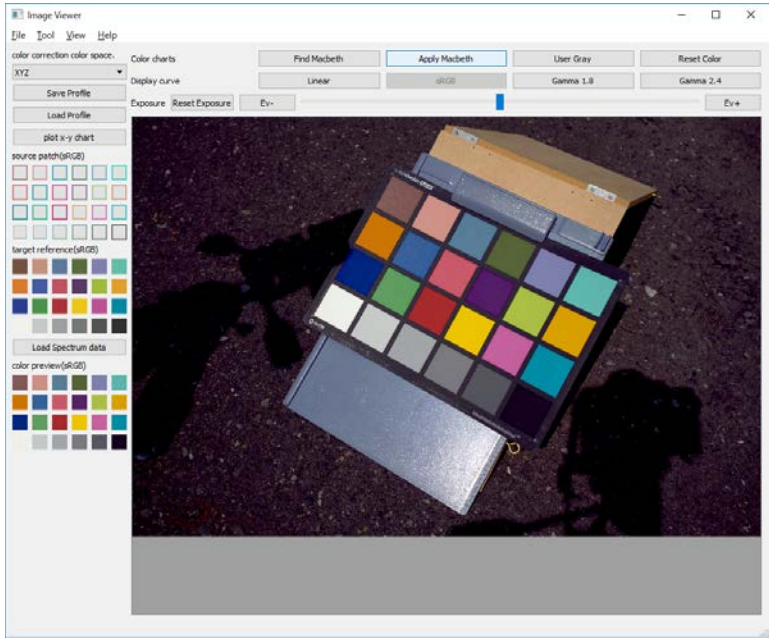
- Reflectance spectrum of Color Checker are precisely measured by Noboru Ohta. [Ohta 1997]
- We also measured reflectance spectrum by ourselves.



Gamut Reverse Engineering

- We can calculate reverse gamut matrix from reflectance spectrum and illumination spectrum using Gauss-Seidel solver.
- See my talk at CEDEC2016(Japanese) for more detail. [Uchimura 2016]





Capturing Device Example

- ILCE-7 series DSLR from SONY.
- Calibrated all the lenses we used.
- Calibrated all the cameras we used.

- That was really tough task.


表参道
神宮前五丁目
大原
Ohara
富ケ谷
Tomigaya
渋谷
Shibuya
413
新宿
Shinjuku
明治通り
RALPH LAUREN
ROADSTER


NO PARKING
TTS


止まれ
一方通行
BMW M4 Coupé

1. Introduction of This Course (5 min.)
2. Introduction of Color and HDR (20 min.)
3. Practical HDR Output Techniques (15 min.)
4. Asset Creation (15 min.)
    a. Capturing HDR
    b. Capturing Wide Color
    c. Handling HDR Images and Videos
5. Implementations and Results in Our Game (30 min.)
6. HDR Working Environment (10 min.)

Textures for light emission

Textures for sky dome

Background images for Scapes (in-game feature)

HDR Image Formats

- We use a few HDR image formats

Asset Creation	Mainly used by artists during asset creation and editing
Runtime	Mainly used by the game itself

HDR Image Formats for Asset Creation

- There are many image formats that support HDR images
- We mainly use two formats:
  - **.hdr**: Radiance HDR / RGBE image format
  - **.exr**: OpenEXR

Radiance HDR / RGB image format (.hdr)

- http://radsite.lbl.gov/radiance/refer/filefmts.pdf
- **Pros**:
  - Easy to understand format specification
  - Many software exist that can handle it
- **Cons**:
  - Can’t store complex data (e.g. multi channels)
  - Not so good precision
    - Shared exponent method
    - It is almost enough for our purposes…
  - Compression method is simple and inefficient

OpenEXR (.exr)

- http://www.openexr.com/
- **Pros**:
  - Can store complex data (deep data)
  - Can store better precision data (16, 32-bit floating point)
  - Can be handled by major DCC software
  - Better compression methods than Radiance HDR
    - Lossy compression method also available
- **Cons**:
  - Lossy compression's quality is not as good as JPEG XR

Why These Two Formats?

- There are other HDR image formats but we are using these formats because of portability and rich features

.hdr (Radiance HDR)	Good portability
.exr (OpenEXR)	Rich features

HDR Image Formats for Runtime

- We also need to use HDR images at runtime
- Runtime HDR image formats requirements
  - **Compact**
    - Needs to be stored in game disk, thus needs a compact size
  - **Fast Decoding**
    - Decoding performance directly impact loading time
  - **GPU friendly**
    - Runtime memory efficiency and cost of sampling textures

BC6H Format for Runtime

- Block compression format
  - Introduced in DirectX11
- **Pros**:
  - Can be decoded very fast
    - Just move from a storage into memory
  - GPU-native compression format
    - Reduce memory consumptions and bandwidth

BC6H Format for Runtime

- Block compression format
  - Introduced in DirectX11
- **Pros**:
  - Can be decoded very fast
    - Just move from a storage into memory
  - GPU-native compression format
    - Reduce memory consumptions and bandwidth

→ We use the BC6H format for HDR textures when performance requirements are severe (e.g. race sequences).

BC6H Format for Runtime

- **Cons**:
    - Compression rate is not so good
        - 1byte / pixel

BC6H Format for Runtime

- **Cons**:
  - Compression rate is not so good
    - 1byte / pixel

→ For Scapes, we need to handle a lot of HDR images, so another format was introduced.

Scapes

"Scapes is an in-game feature that allows you to position a car in scenes from around the world and take photos."

Scapes Background Images

Formats	**HDR**
Resolution	**Very high (between 6000x4000 and 8000x5000)**
Numbers	**Over 1,000**

Scapes Background Images

Formats	**HDR**
Resolution	**Very high (between 6000x4000 and 8000x5000)**
Numbers	**Over 1,000**

→ BC6H is inefficient! The estimated total size was over 30GB (Half the size of a game disk ☹) and the file size was a large issue.

JPEG XR

- https://jpeg.org/jpegxr
- Developed by Microsoft
- **Pros**:
  - **High quality and efficient lossy algorithm**
  - Can store complex data
  - Supports not only HDR format (like fp32, fp16) but many other color formats
  - Reference implementation is open and under the public domain
- **Cons**:
  - There are few software that supports JPEG XR appropriately
    - But the image preview in Win10 supports JPEG XR

Our Implementation of JPEG XR

- Based on a public domain version of JPEG XR
- Improved the decoding performance
  - Code optimizations using SIMD
  - Parallel decoding features
  - About x10~x20 faster than the original code (PlayStation®4) ☺

Our Implementation of JPEG XR

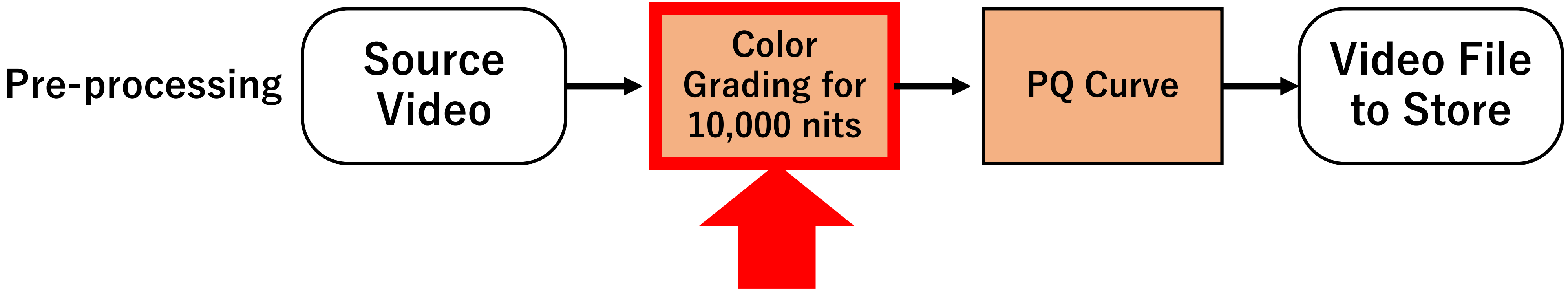
- Based on a public domain version of JPEG XR
- Improved the decoding performance
  - Code optimizations using SIMD
  - Parallel decoding features
  - About x10~x20 faster than the original code (PlayStation®4) ☺

- On average, compared to BC6H, the disk consumption decrease from about 1/2 to 1/3 of the size ☺

➡ We use this modified JPEG XR format for Scapes background images and in-game photos captured by users.

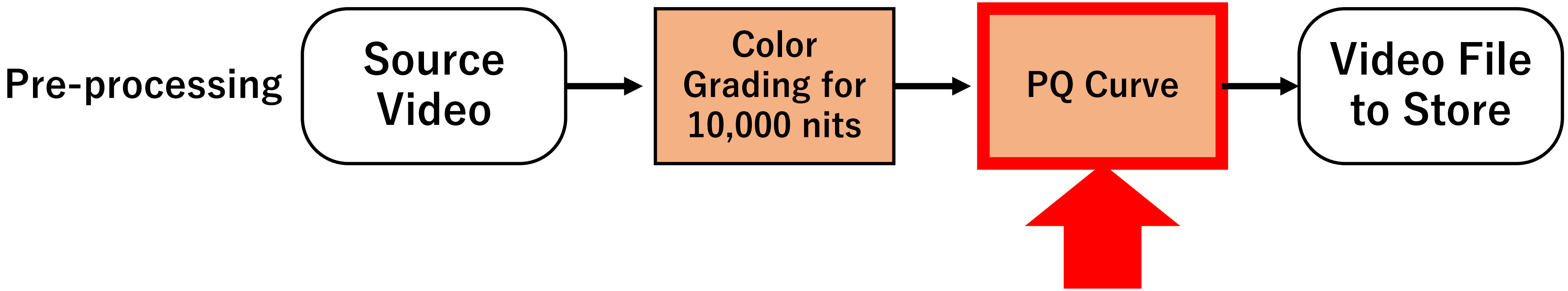
HDR Video Formats

- **Pre-processing**
  - In-game videos are graded for 10,000 nits targets
    - e.g. opening and ending videos
  - Manually graded to expand the dynamic range



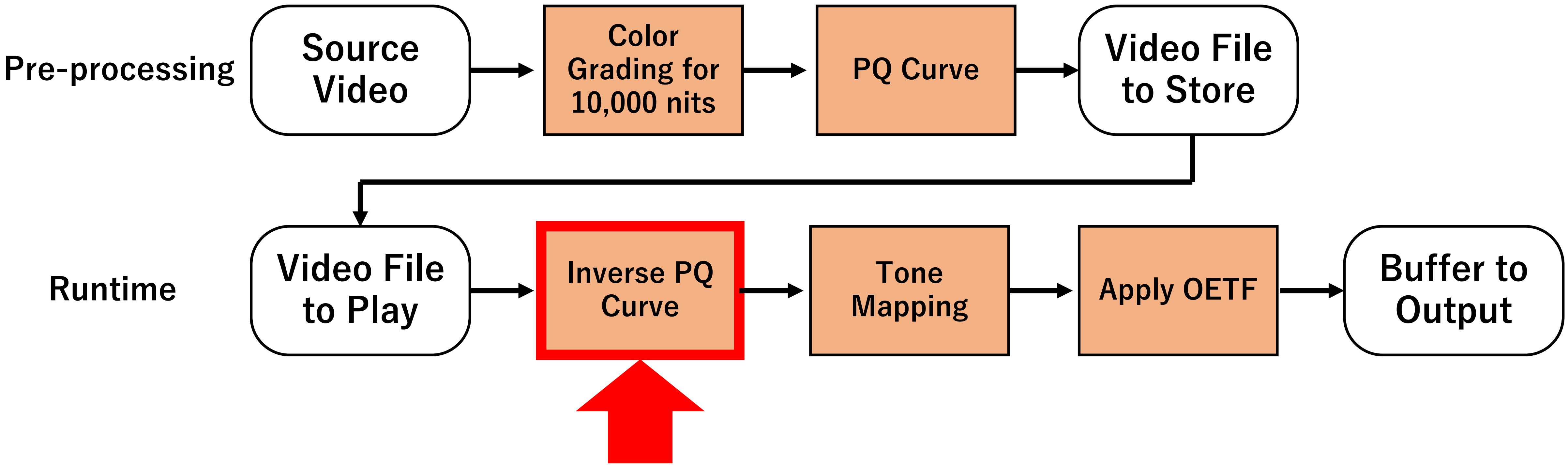
HDR Video Formats

- **Pre-processing**
  - Stored as a file after PQ curve is applied



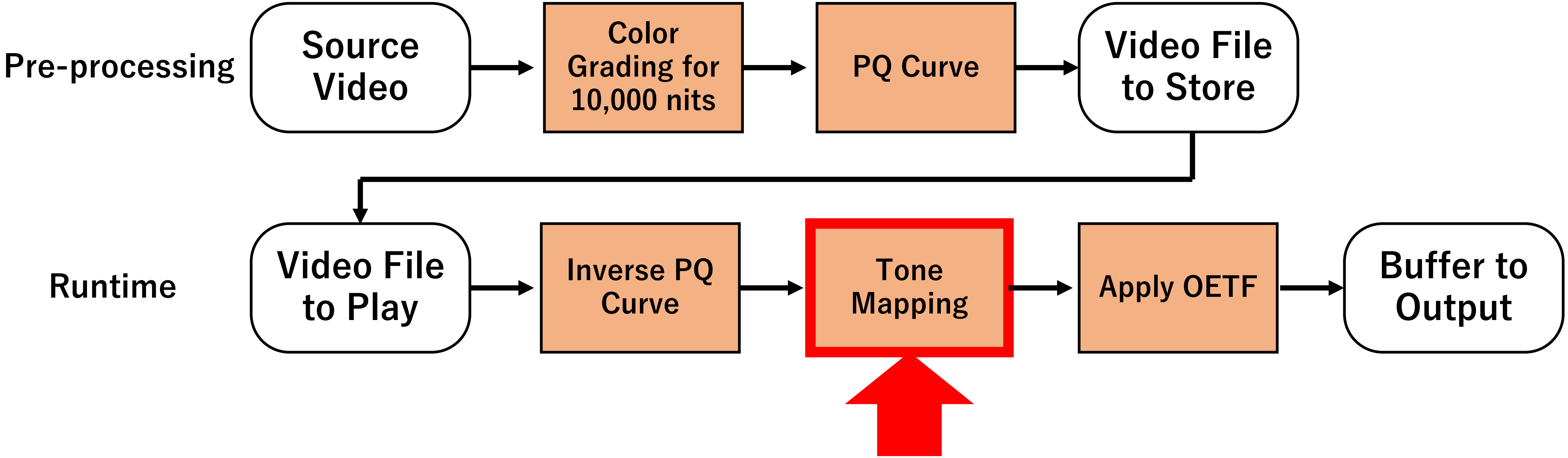
HDR Video Formats

- **Runtime**
  - Inverse PQ curve is applied and then the buffer is linearized



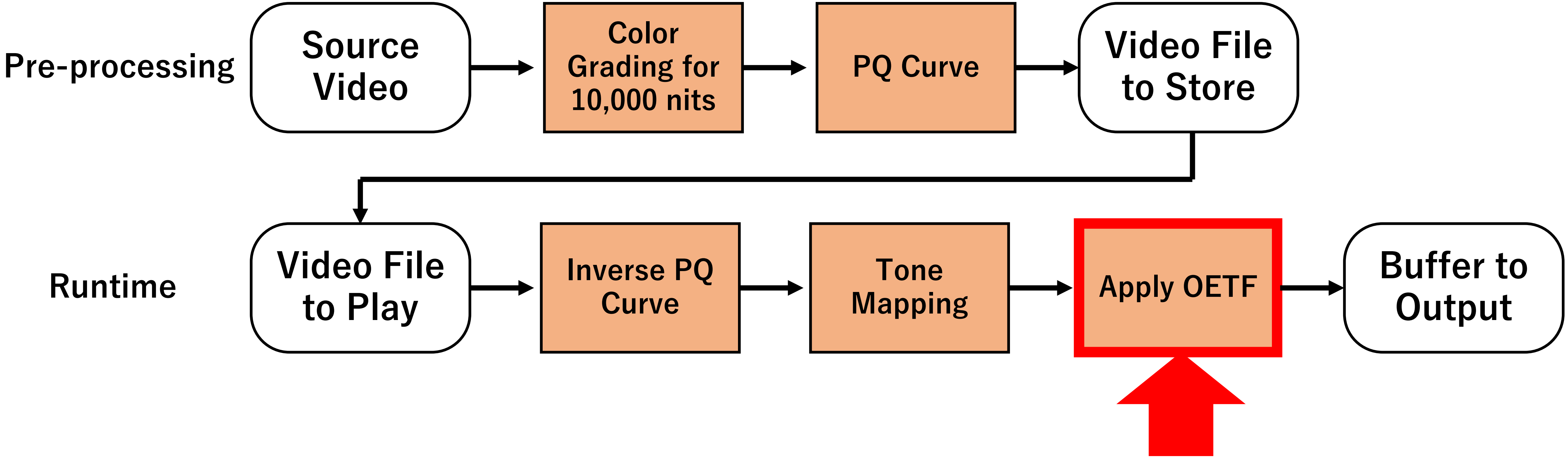
HDR Video Formats

- **Runtime**
  - GT tone mapping is applied for each output device



HDR Video Formats

- **Runtime**
  - OETF is applied for each output device



HDR Video Formats

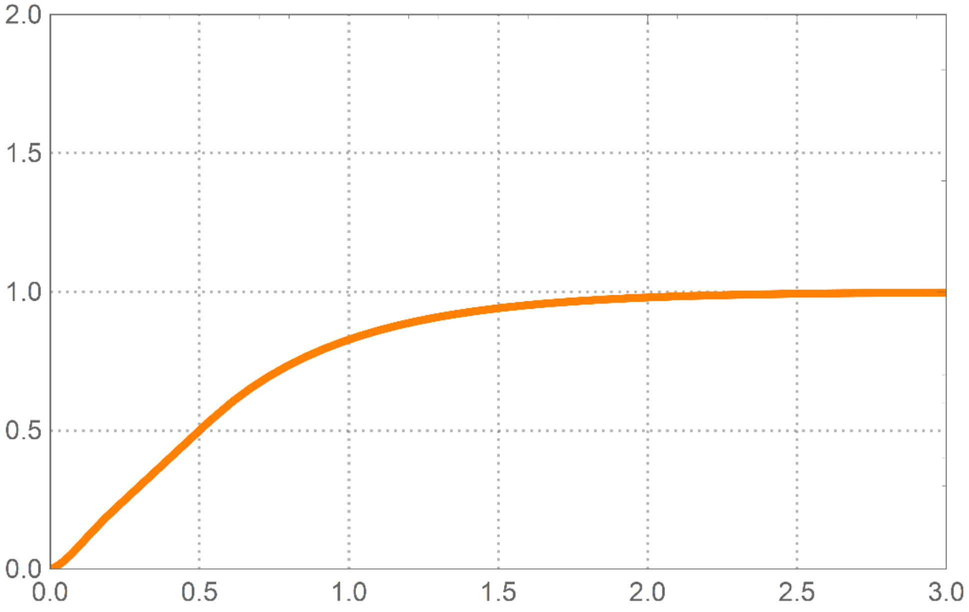
- HEVC format is used at runtime
- **Pros**:
  - Can store 10 bit format pixels
  - Good quality (better than H.264)
- **Cons**:
  - Expensive to encode and decode
  - High licensing fee

1. Introduction of This Course (5 min.)
2. Introduction of Color and HDR (20 min.)
3. Practical HDR Output Techniques (15 min.)
4. Asset Creation (15 min.)
5. Implementations and Results in Our Game (30 min.)
6. HDR Working Environment (10 min.)

1. Introduction of This Course (5 min.)
2. Introduction of Color and HDR (20 min.)
3. Practical HDR Output Techniques (15 min.)
4. Asset Creation (15 min.)
5. Implementations and Results in Our Game (30 min.)
    a. Implementation of GT Tone Mapping for HDR Output
    b. Tone Mapping Calibration for Output Devices
    c. Rendering in a Wide Color Space and HDR
    d. HDR Color Grading
6. HDR Working Environment (10 min.)

Implementation of GT Tone Mapping for HDR Output

- Our runtime tone mapping method is “GT Tone Mapping”
  - We already explained the theory behind it in this course
  - How did we implement it in our game?



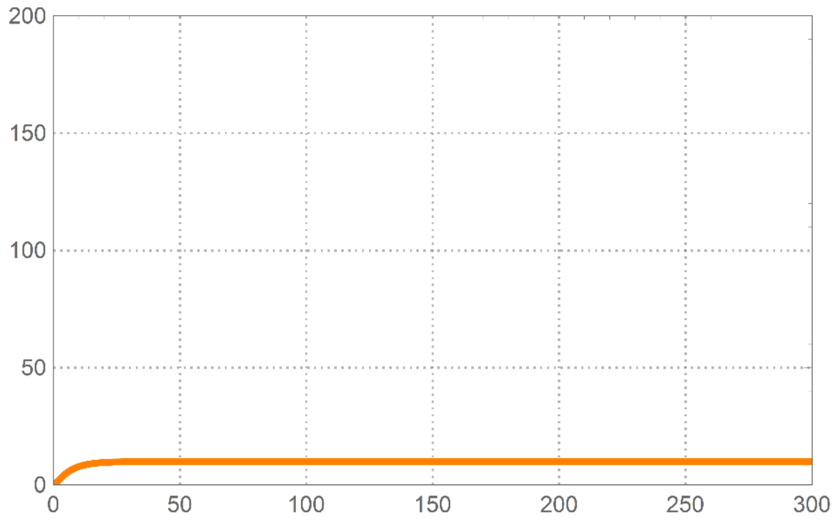
Performance Issue

- The simple implementation has a high computational cost
  - Complex function (pow and exp functions are heavy on the GPU)

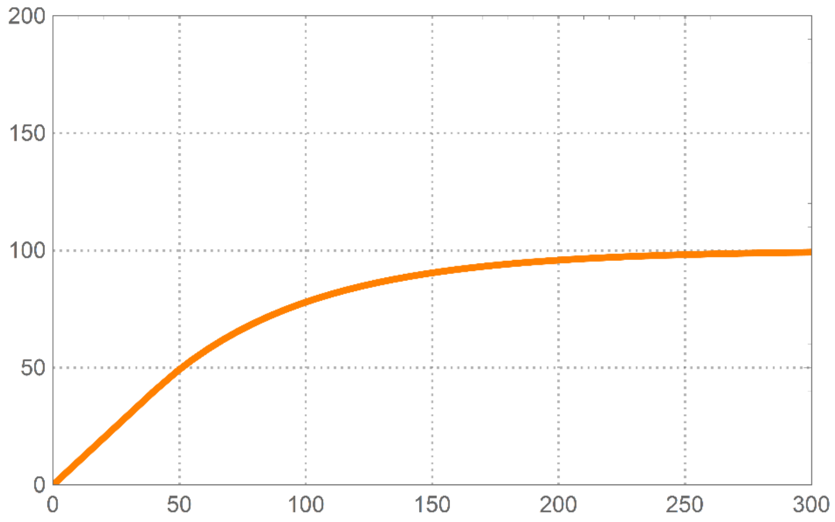
```
float GTTonemap(float x)
{
    float L = m + a * (x - m);
    float T = m * pow(x/m, c);
    float S = P - (P - S1) * exp(-C2*(x - S0)/P);
    float w0 = 1 - smoothstep(0.0f, m, x);
    float w2 = (x < m+l)?0:1;
    float w1 = 1 - w0 - w2;
    return (float)(T * w0 + L * w1 + S * w2);
}
```

Performance Issue

- Approximations (e.g. rational function fitting) need to be calculated for each tone mapping parameter
  - GT tone mapping is "variable" with respect to a device peak brightness



Tone Mapping for
Peak brightness: 1,000 nits

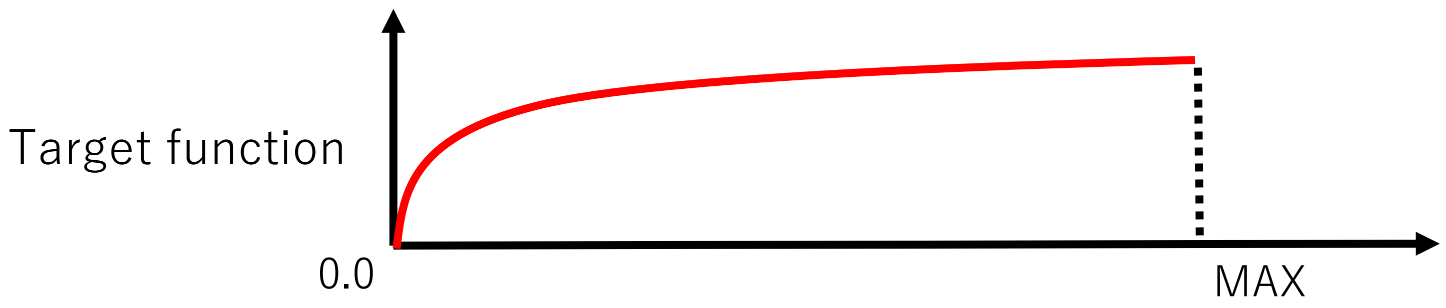


Tone Mapping for
Peak brightness: 10,000 nits

Lookup table (LUT) approach

- We used a lookup table for computing the GT Tone Mapping at runtime
- **Pros**:
  - Approximation error can be reduced by just using a larger size table
  - Cost of sampling lookup table tend to be smaller than the analytic calculation
    - ALU pressure was the main bottleneck in our case
- **Cons**:
  - Memory consumption is larger than using an analytic function
  - Quantization error and discretization error

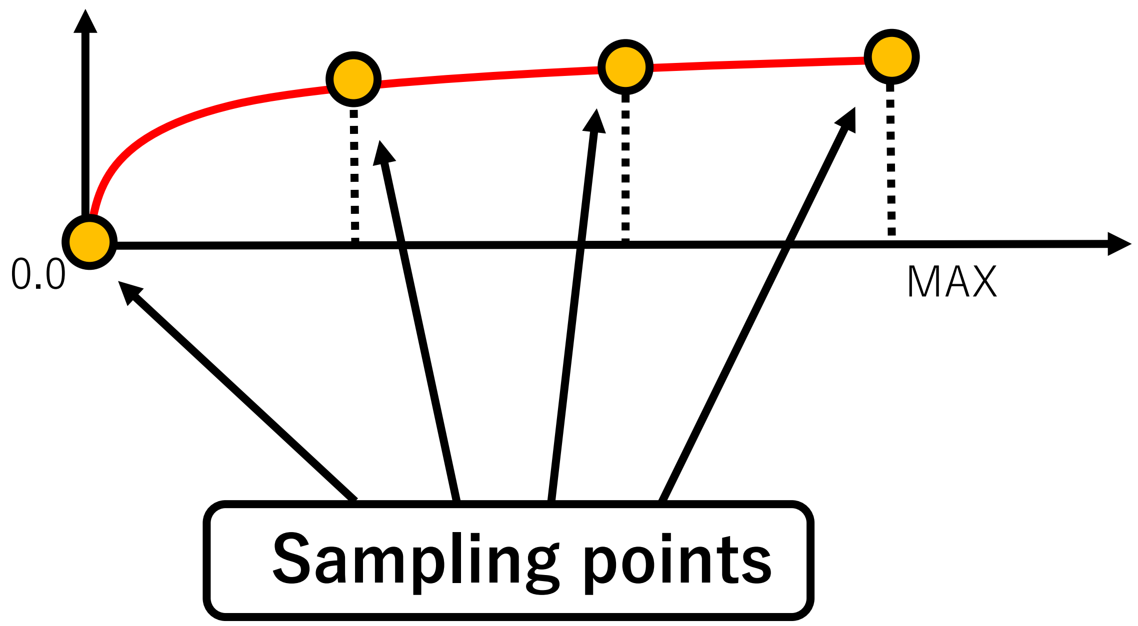
Usual Linear LUT Mapping



Usual Linear LUT Mapping

- Uniformly sample a target function

Target function

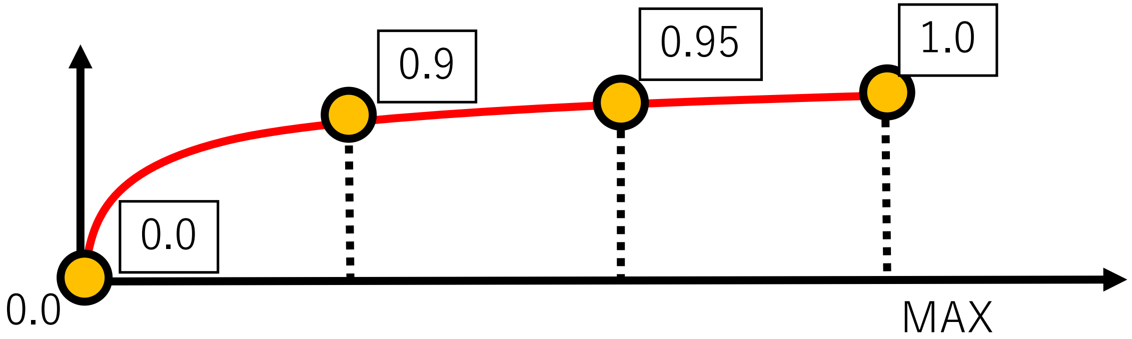



```
sampling_point = lut_index / (N-1) * MAX
```

Usual Linear LUT Mapping

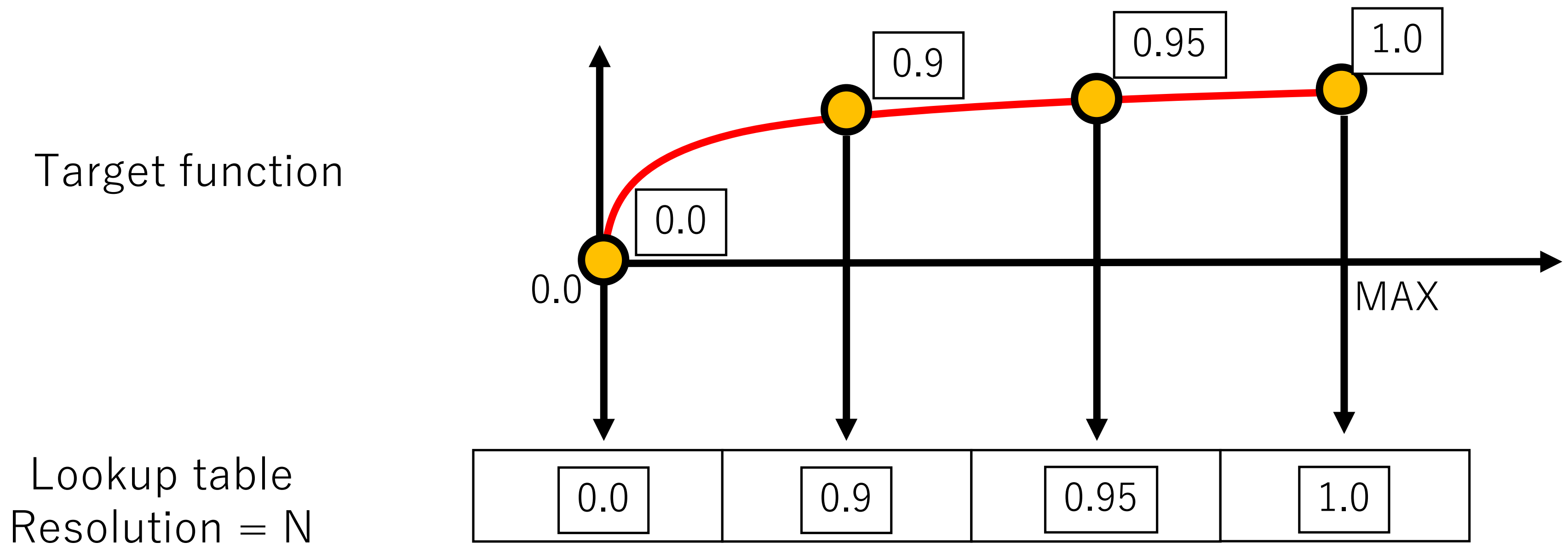
- Uniformly sample a target function

Target function



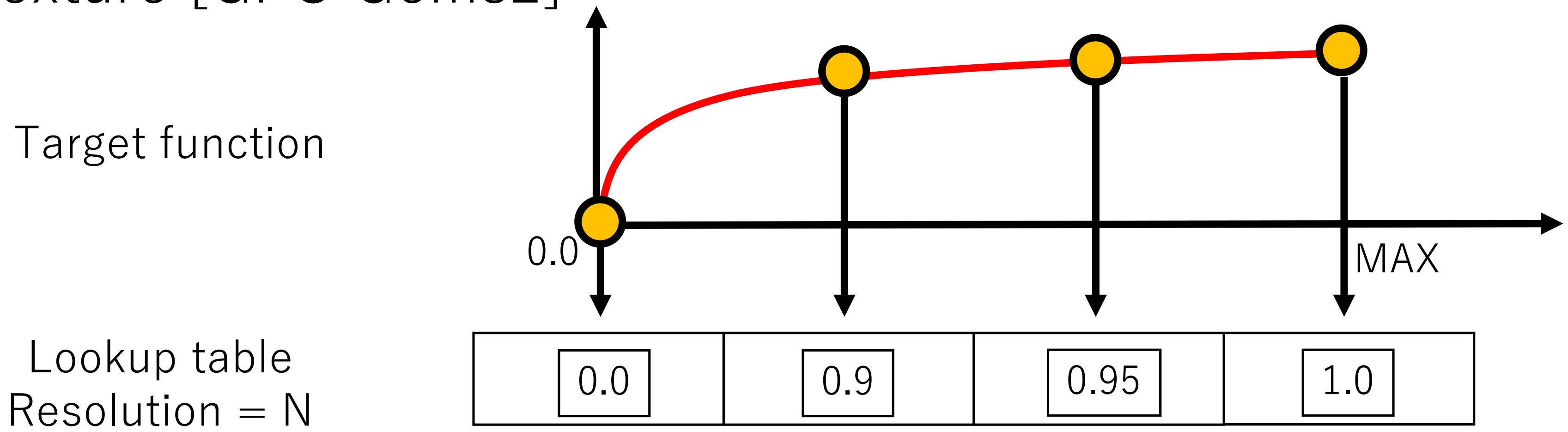
Usual Linear LUT Mapping

- Store sampled values onto a lookup table



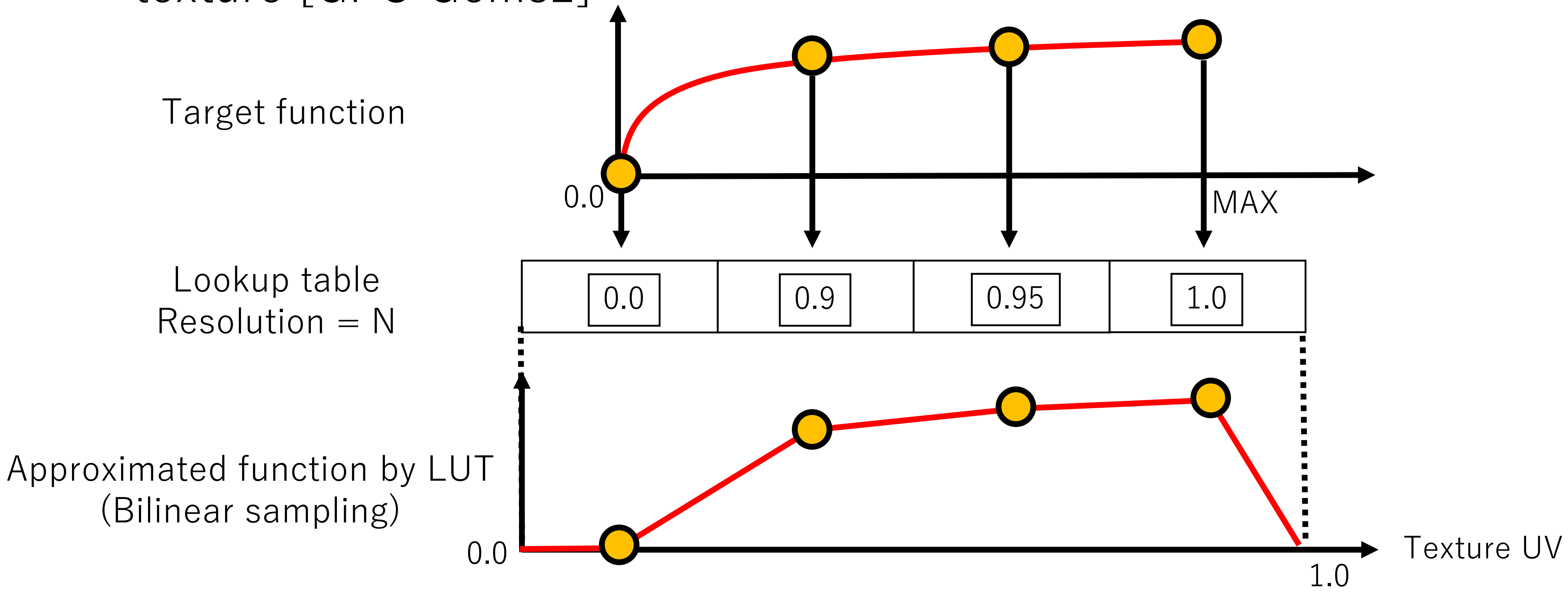
Usual Linear LUT Mapping

- Need to adjust offsets and scale LUT coordinate when sampling LUT texture [GPU Gems2]



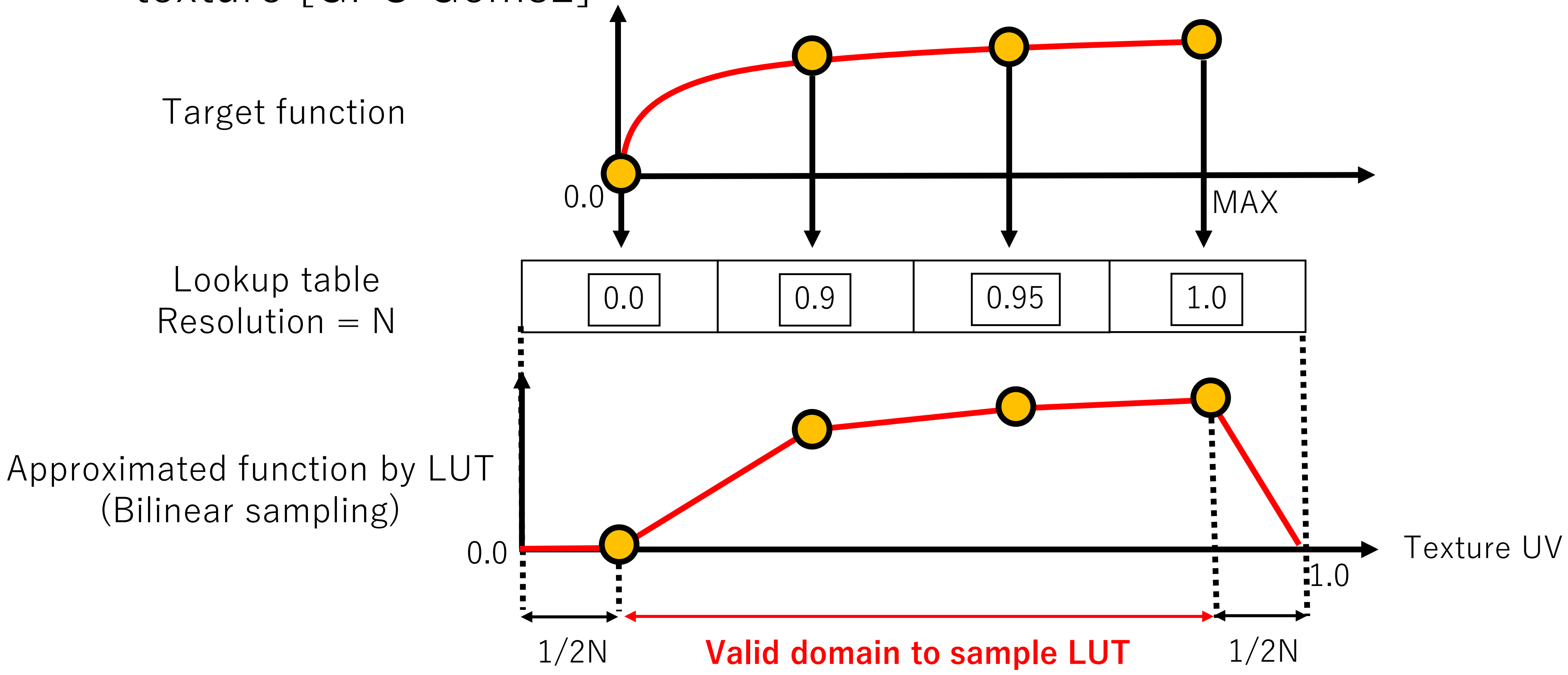
Usual Linear LUT Mapping

- Need to adjust offsets and scale LUT coordinate when sampling LUT texture [GPU Gems2]



Usual Linear LUT Mapping

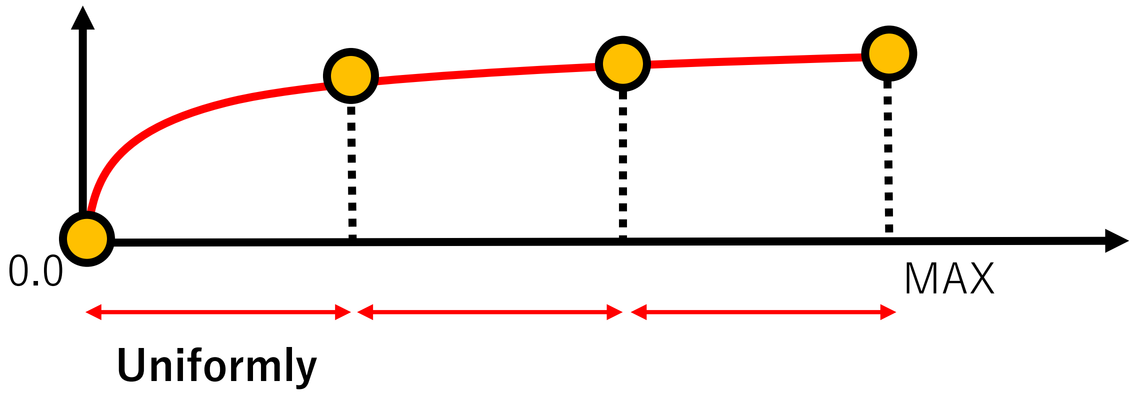
- Need to adjust offsets and scale LUT coordinate when sampling LUT texture [GPU Gems2]



Nonlinear LUT Mapping

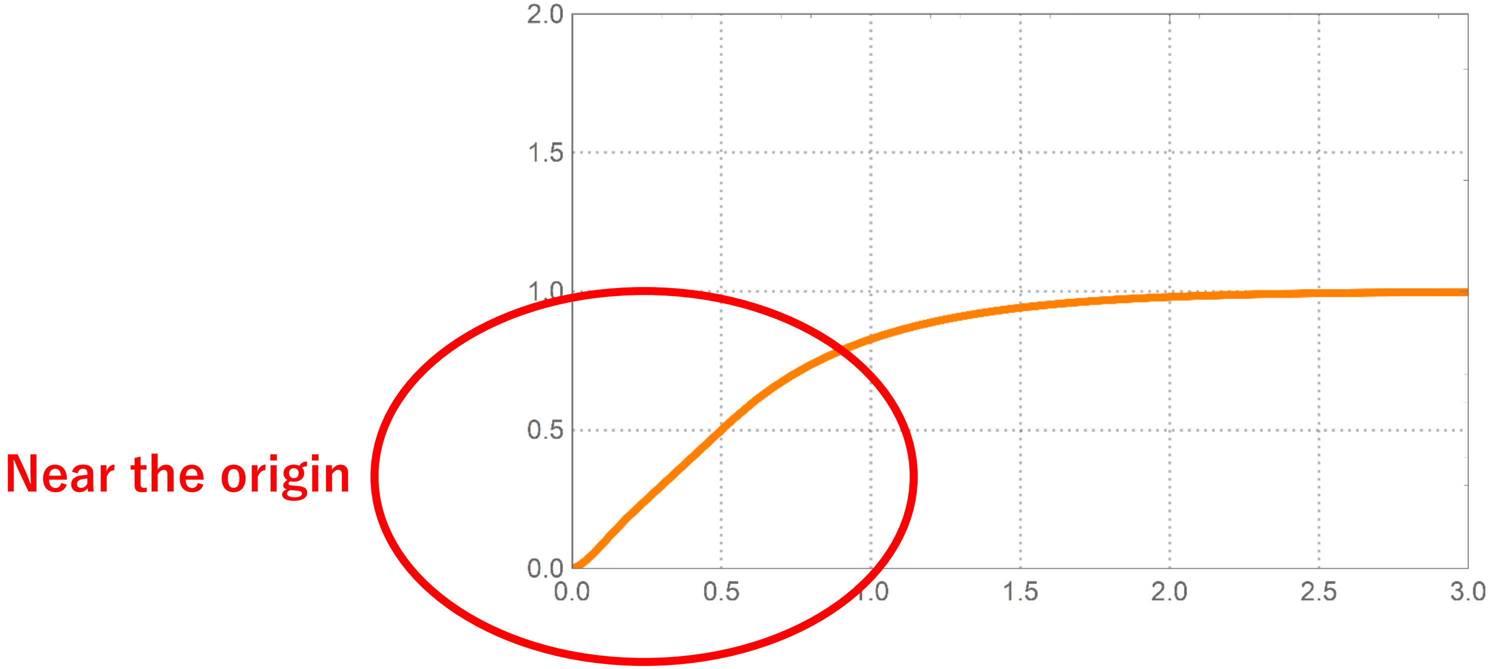
- **NOT** uniformly map the input value to the lookup table coordinates (X-axis)

Target function



Nonlinear LUT Mapping

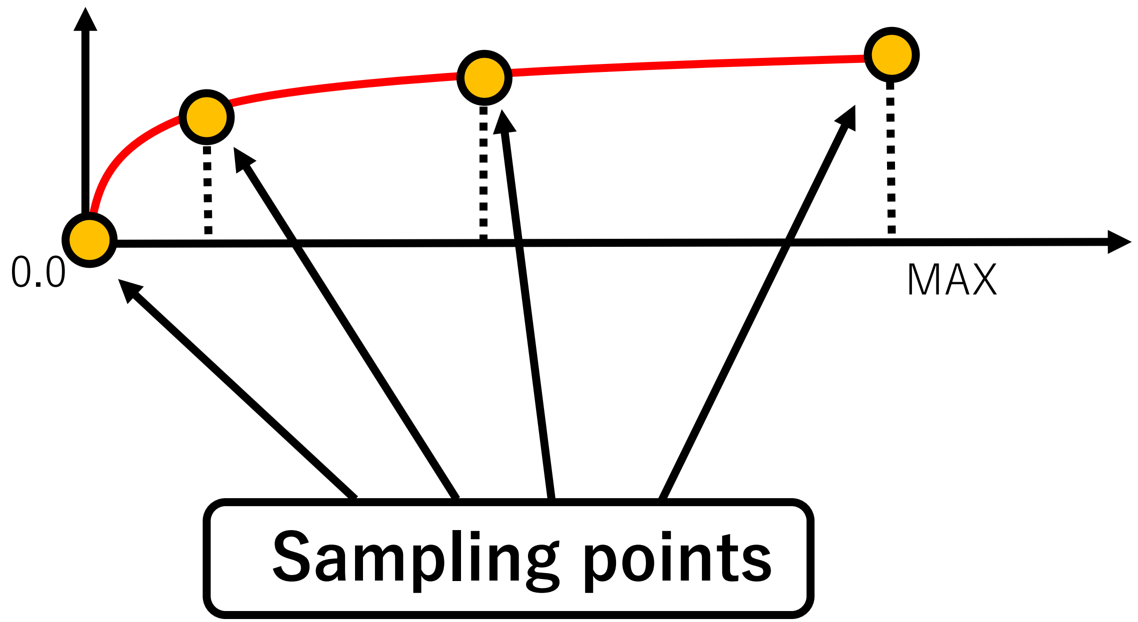
- Accuracy is more important **near the origin**
  - The low part and middle part of the tone mapping function
- So, want to assign more texels near the origin (and less texels far from the origin)



Nonlinear LUT Mapping

- Non-uniformly sample the target function to assign more texels near the origin

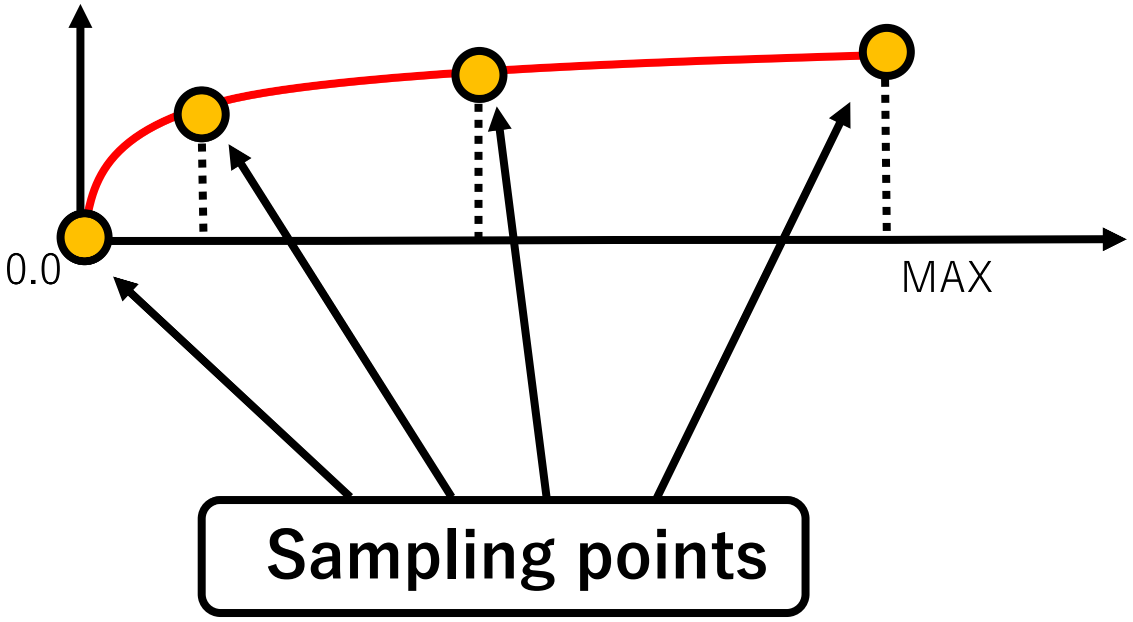
Target function



Nonlinear LUT Mapping

- We use a x^2 mapping so that we can use square root at runtime
  - Other mapping can be used [Malin 2018]

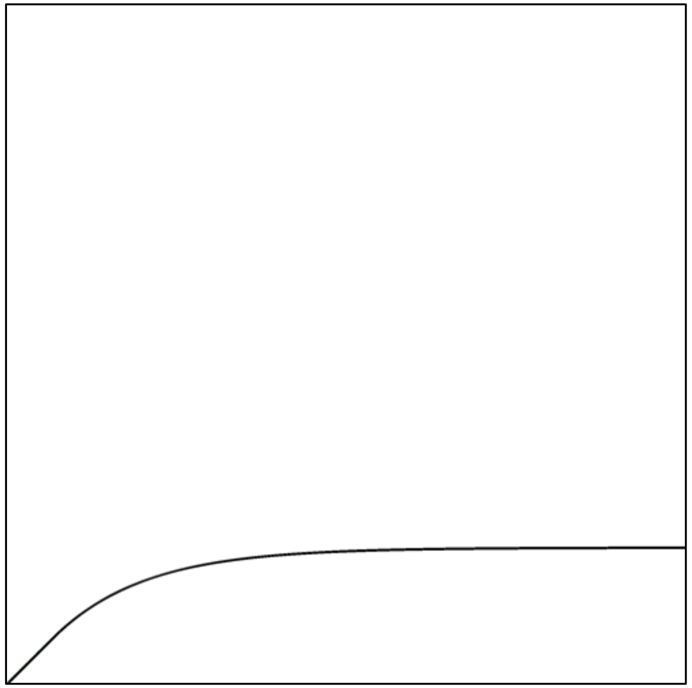
Target function




```
sampling_point = pow(lut_index / (N-1), 2.0f) * MAX
```

Visualization of LUT Sampling Points

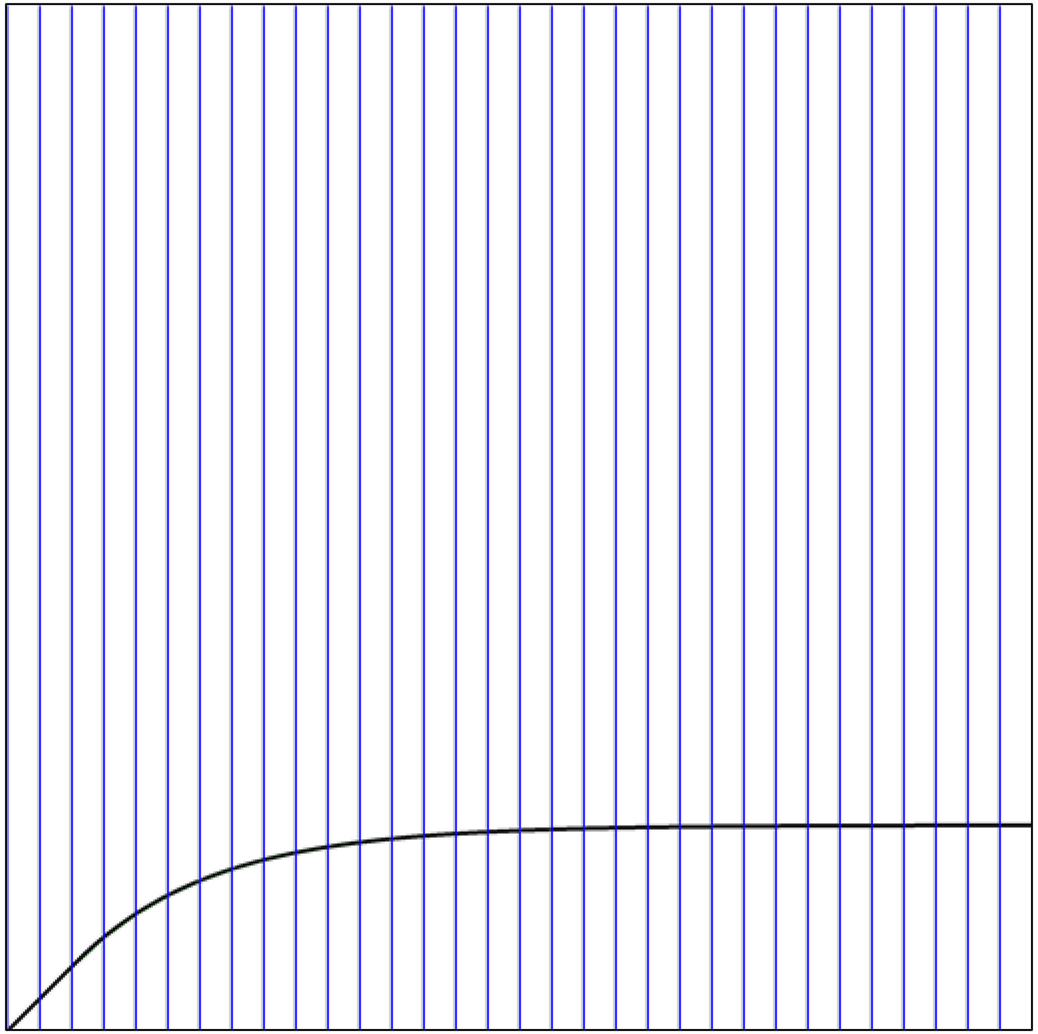
- Whole input range: [0, 50000] nits
- LUT resolution: 32



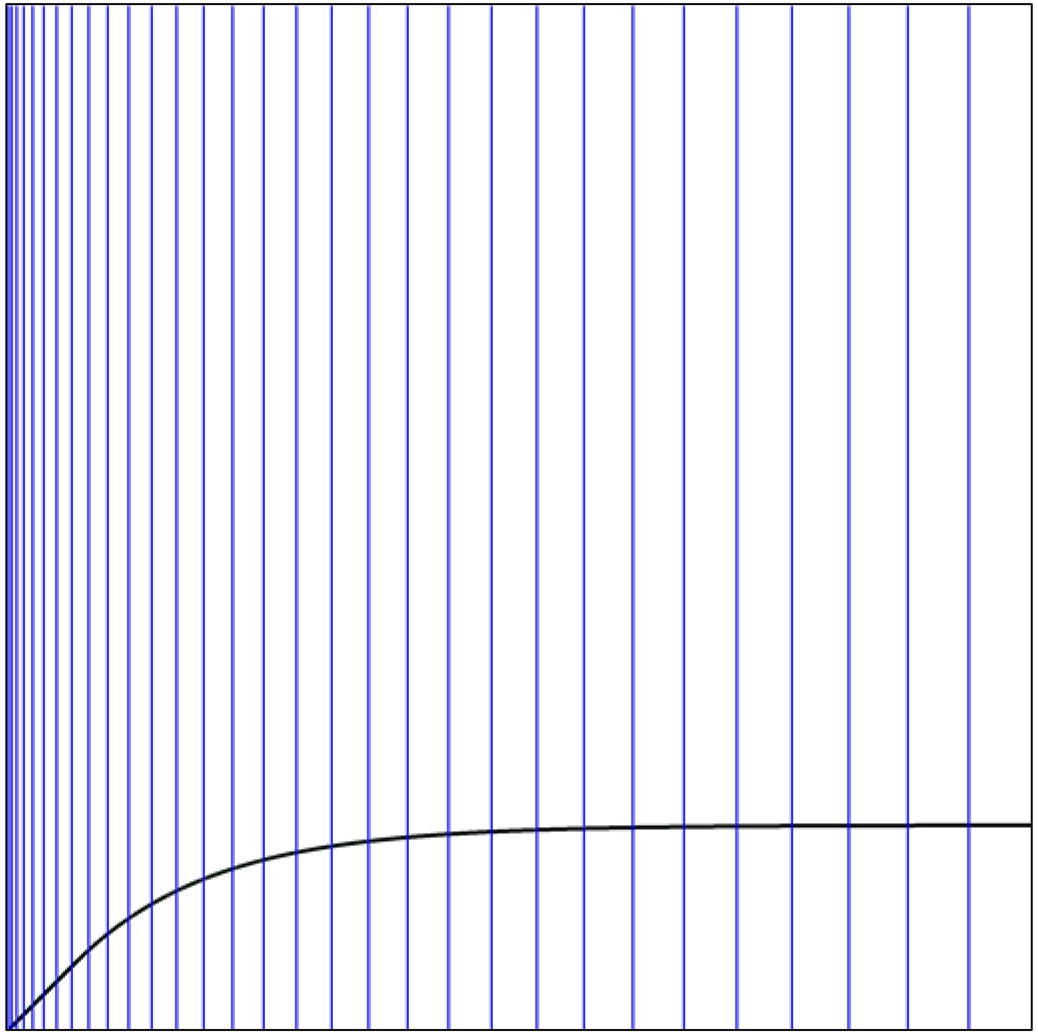
The whole tone mapping function
Range: [0, 50000] nits

Visualization of LUT Sampling Points

Vertical lines:
Sampling points



Linear mapping

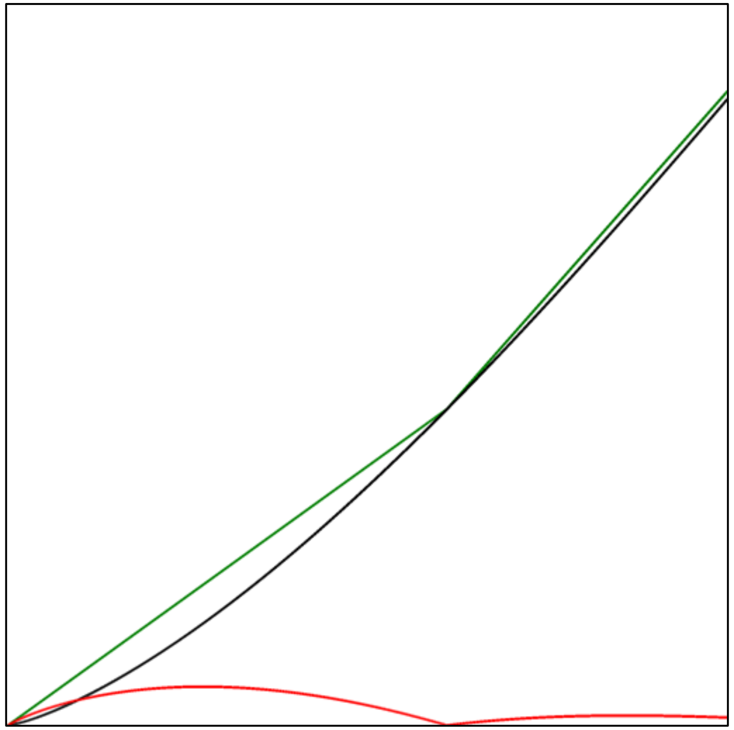


Our nonlinear mapping

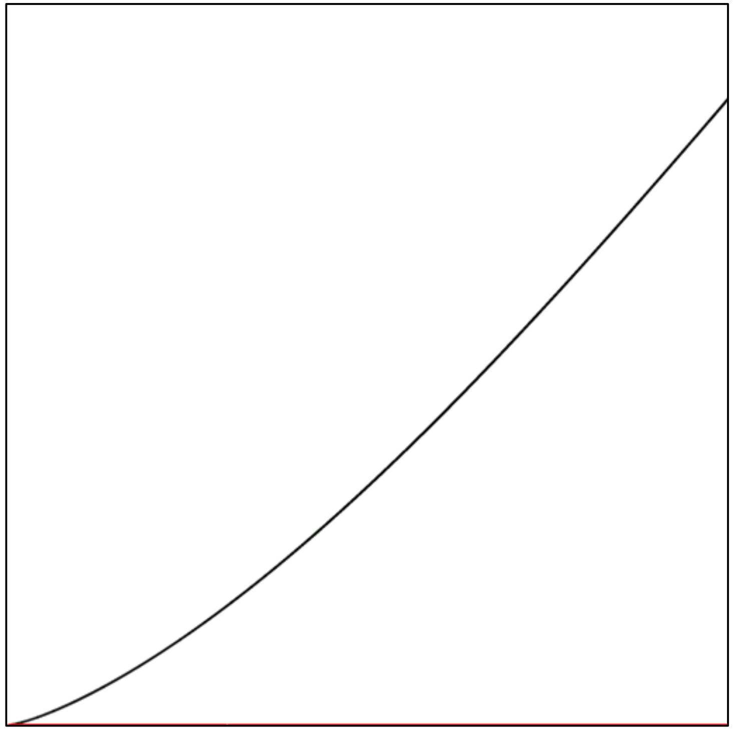
Range: [0, 50000] nits

Comparison

- Comparison of the accuracy between linear mapping and our mapping



Linear mapping



Our nonlinear mapping

Black line: original function
Green line: LUT approximation
Red line: root squared error
(LUT resolution: 8192)

Range: [0, 10] nits

Results

- Specification of lookup table in our game

Texture	1D texture
Format	32 bit float
Width	8,192
Total Size	32 KB

- Performance results (PlayStation®4, 1920x1080)

Analytic Calculation	0.40 ms
Our LUT Approach	**0.24 ms**

Our LUT approach includes creating LUT in the compute shader

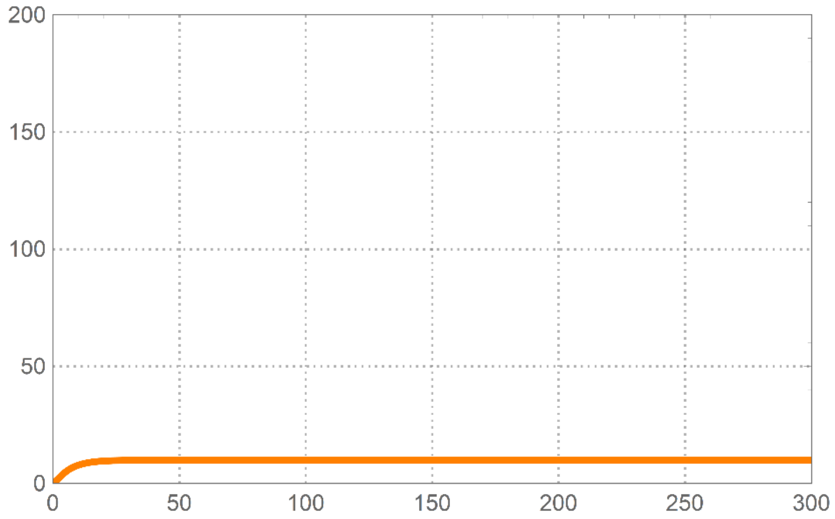
1. Introduction of This Course (5 min.)
2. Introduction of Color and HDR (20 min.)
3. Practical HDR Output Techniques (15 min.)
4. Asset Creation (15 min.)
5. Implementations and Results in Our Game (30 min.)
    a. Implementation of GT Tone Mapping for HDR Output
    b. Tone Mapping Calibration for Output Devices
    c. Rendering in a Wide Color Space and HDR
    d. HDR Color Grading
6. HDR Working Environment (10 min.)

Tone Mapping Calibration for Output Devices

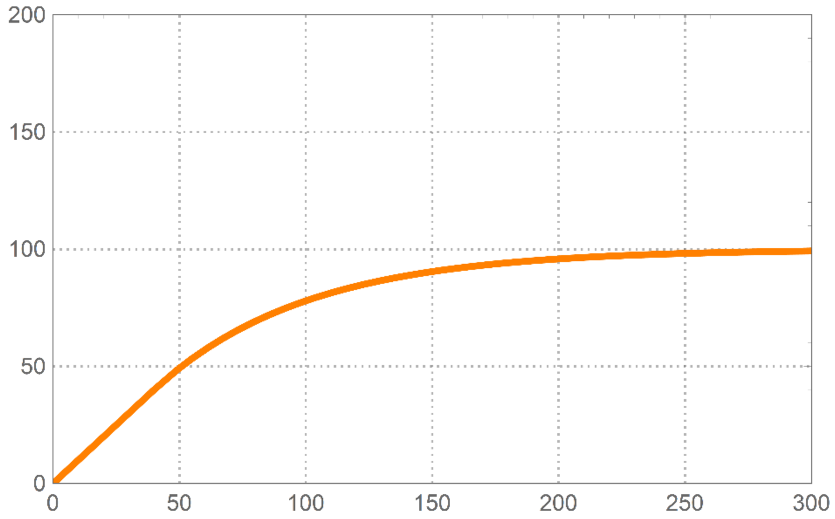
- Appropriate HDR output needs information about output HDR devices
  - Needs information about “peak brightness” of a device

Tone Mapping Calibration for Output Devices

- Tone mapping uses this parameter to adapt rendered images to output devices
  - Want to output optimal images adapted to the dynamic range of the output device



Tone Mapping for
Peak brightness: 1,000 nits



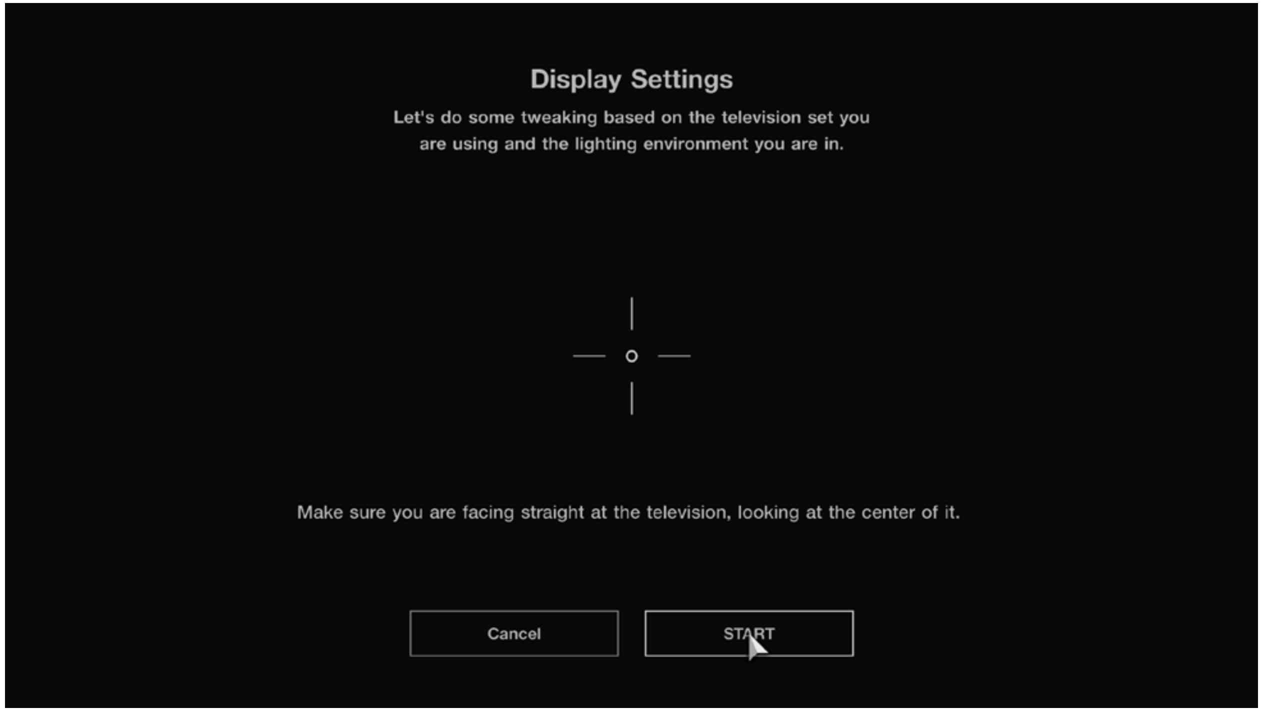
Tone Mapping for
Peak brightness: 10,000 nits

Calibrations

- Want to know a peak brightness, so measure!
- Recent games often have their own calibration processes
- When starting games for the first time, users need to calibrate their screens

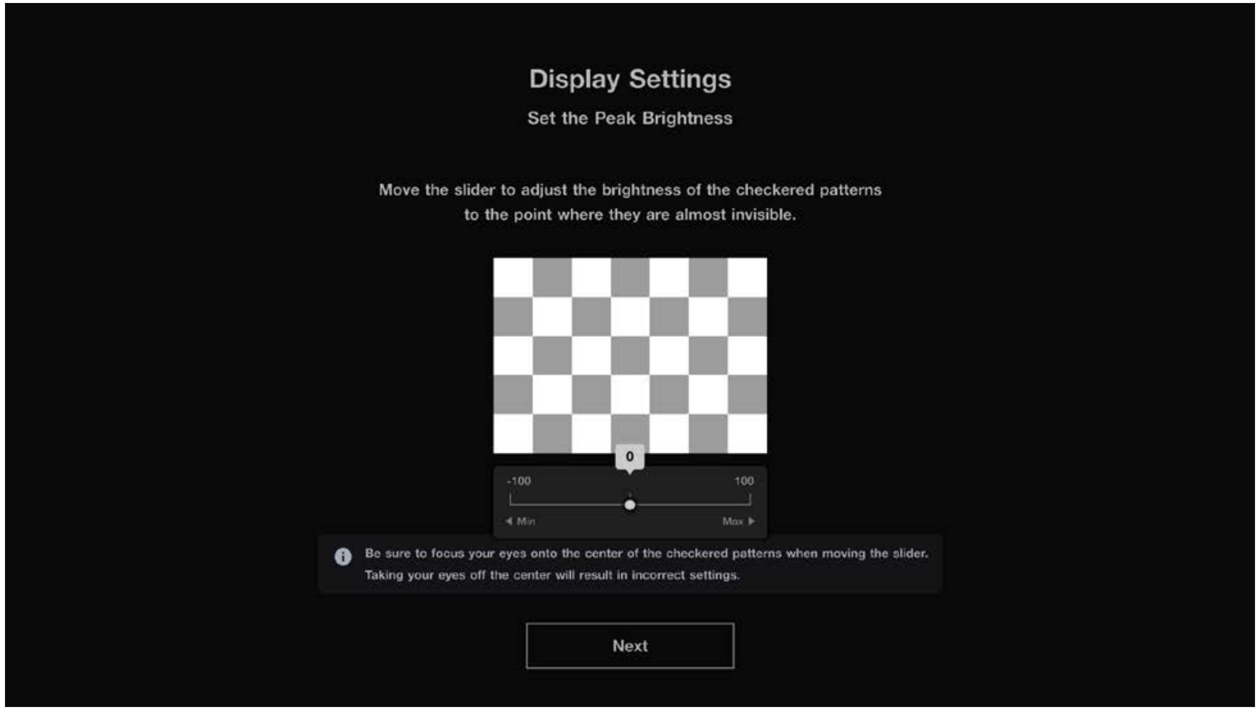
Our Calibration Process (1/2)

- First, make sure users are facing straight at the device
  - Particularly in HDR devices, viewing angle affects the visual appearance greatly



Our Calibration Process (2/2)

- Second, move the slider to adjust the brightness of the checker board patterns to the point where they are almost invisible



Display Settings

Set the Peak Brightness

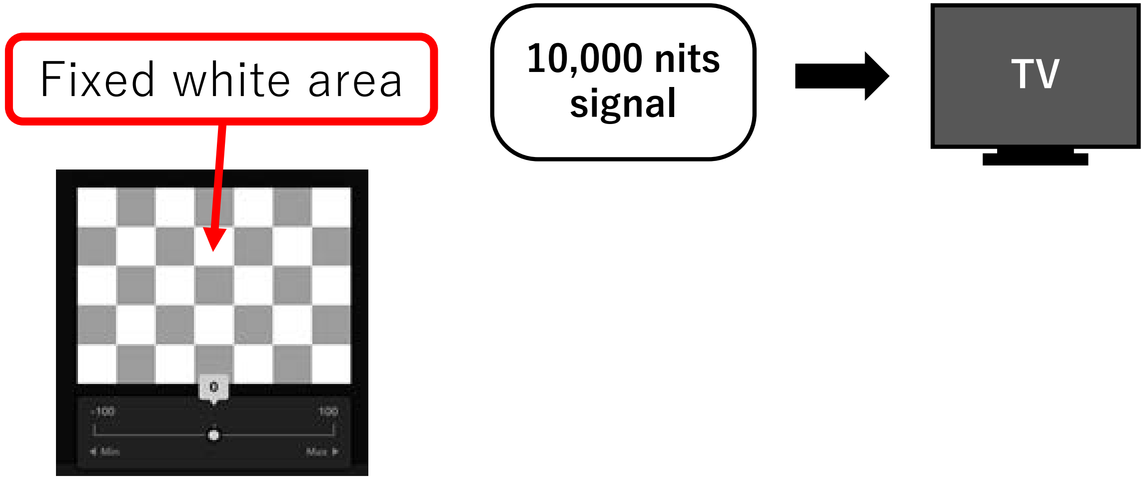
Move the slider to adjust the brightness of the checkered patterns
to the point where they are almost invisible.

ⓘ Be sure to focus your eyes onto the center of the checkered patterns when moving the slider.
Taking your eyes off the center will result in incorrect settings.

Next

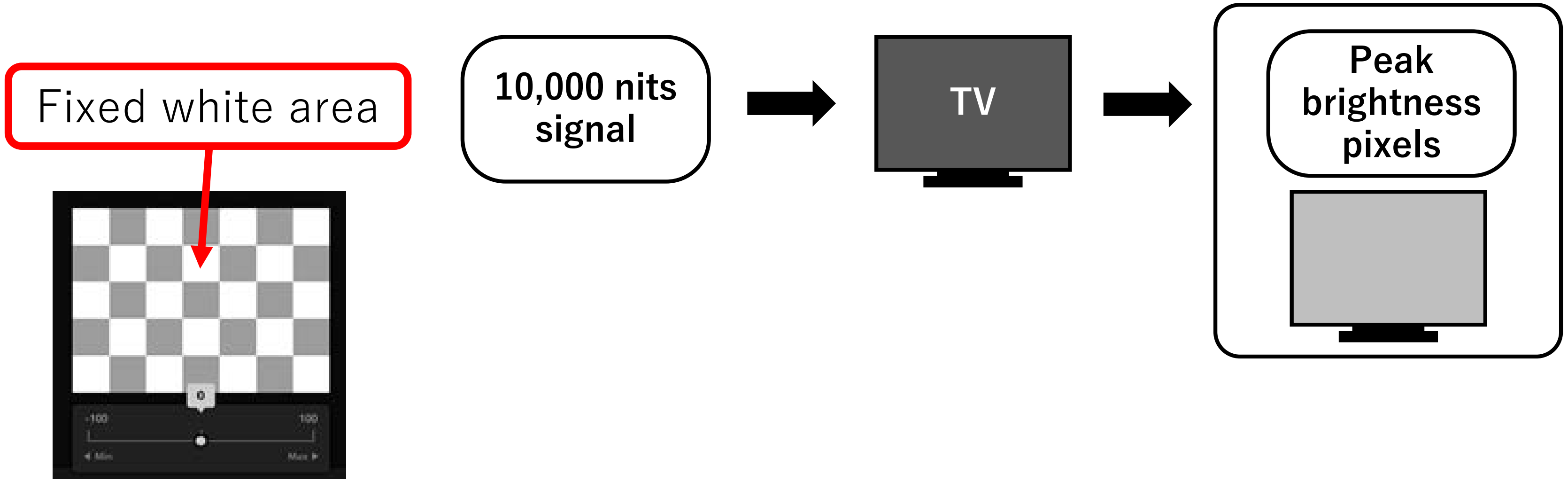
How Can This Estimate the Peak Brightness?

- In the fixed white area
  - The input signal at the device is **10,000 nits** so the device tries to output **10,000 nits pixels**



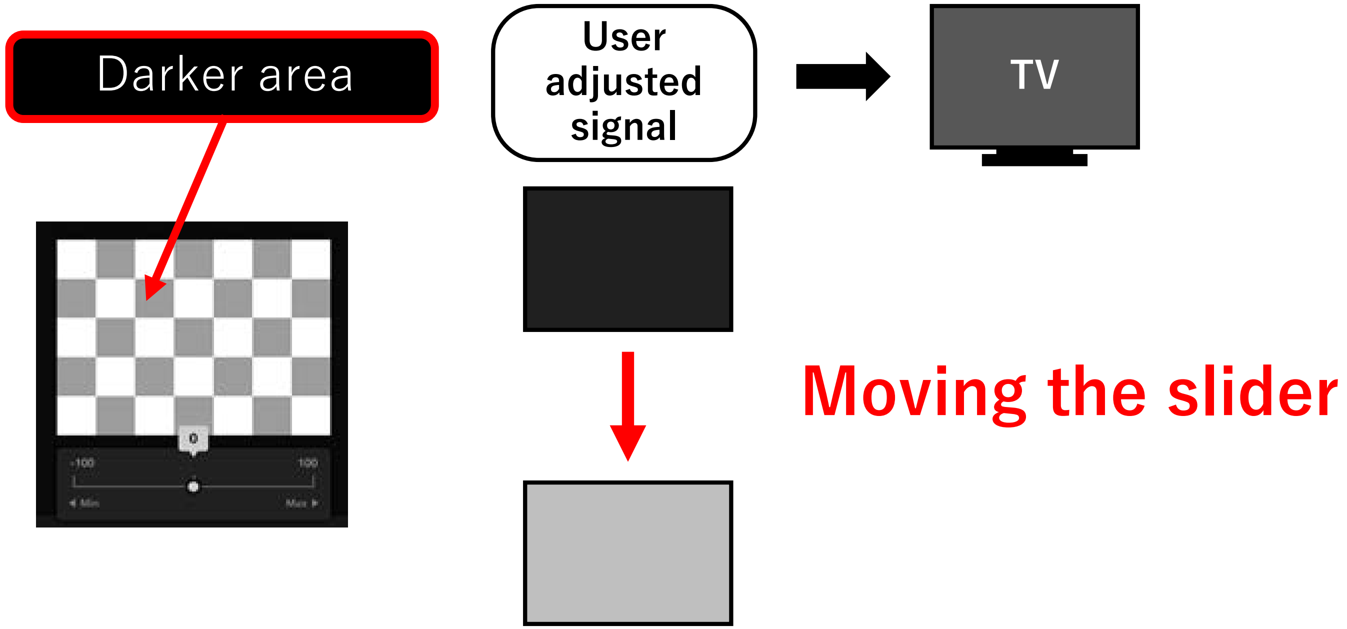
How Can This Estimate the Peak Brightness?

- In the fixed white area
  - However, the typical device can’t output such bright pixels, so this is actually the output device’s peak brightness



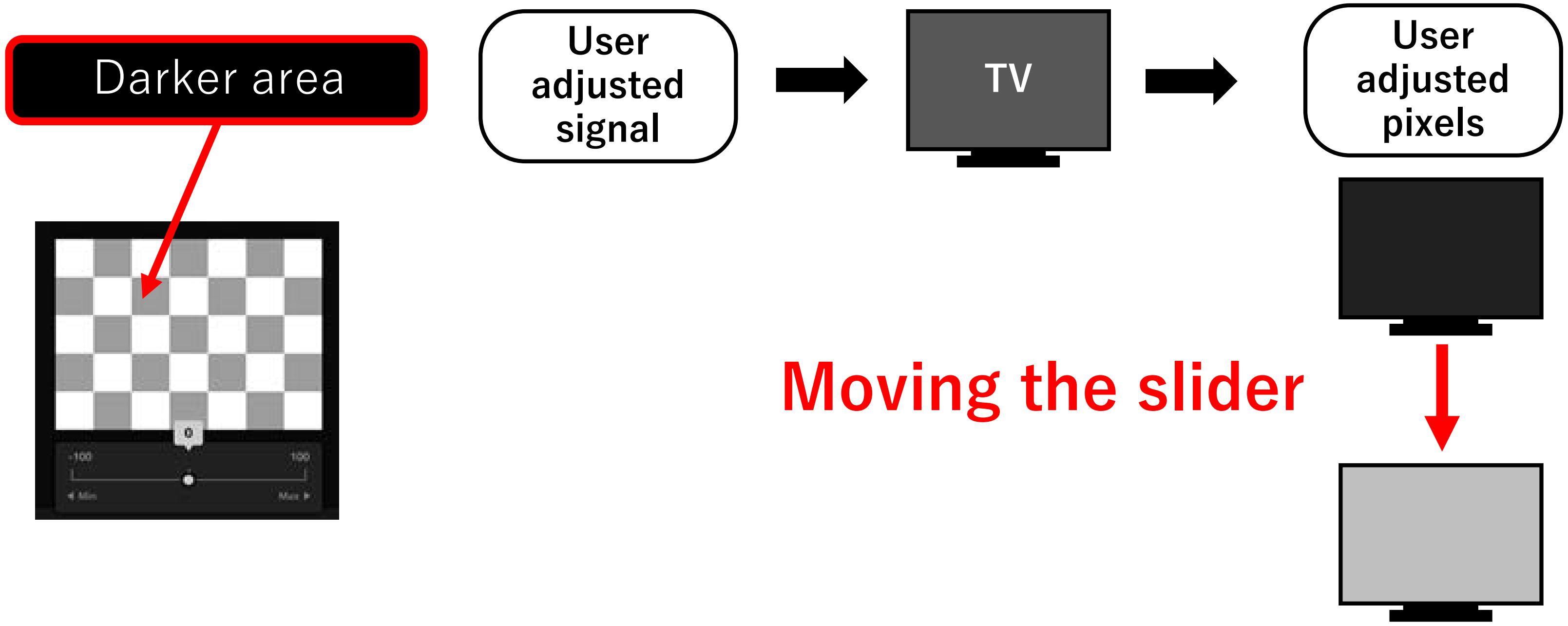
How Can This Estimate the Peak Brightness?

- In the darker area
  - When moving the slider, the input signal is adjusted and brightened



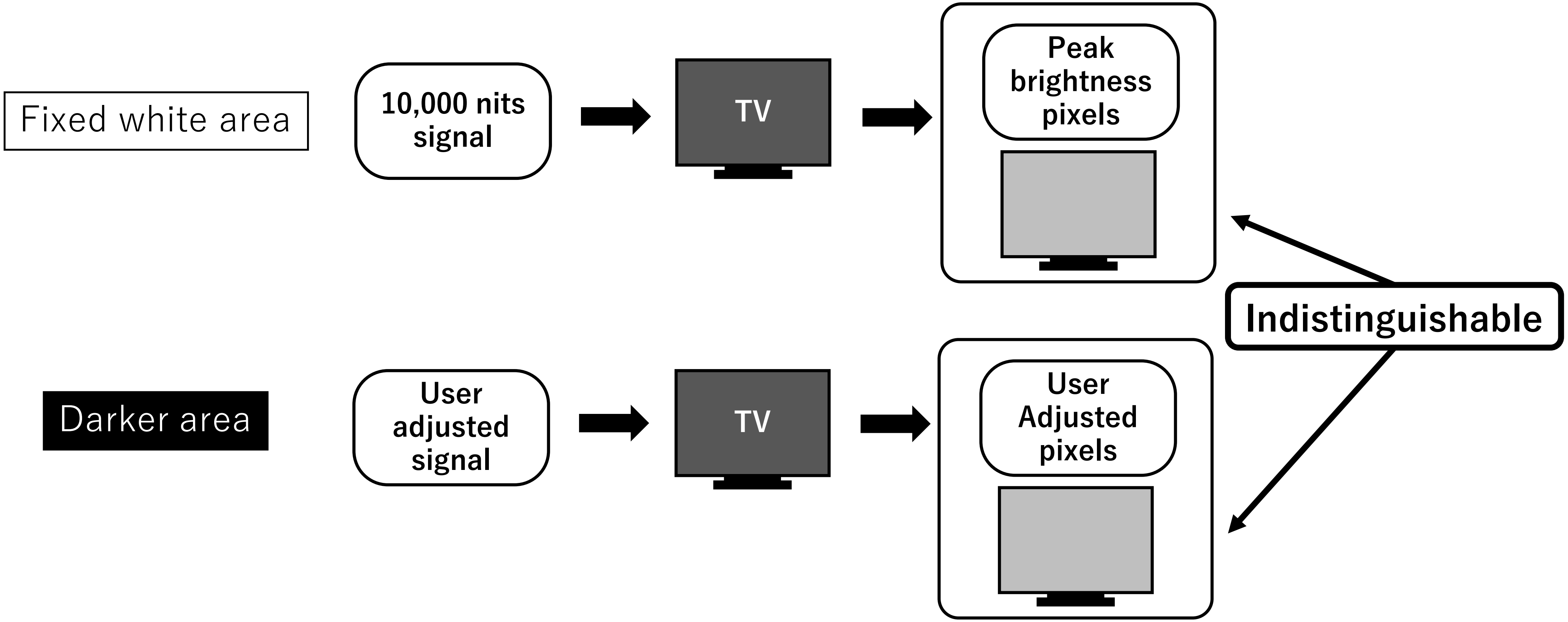
How Can This Estimate the Peak Brightness?

- In the darker area
  - The output pixels also gets brighter



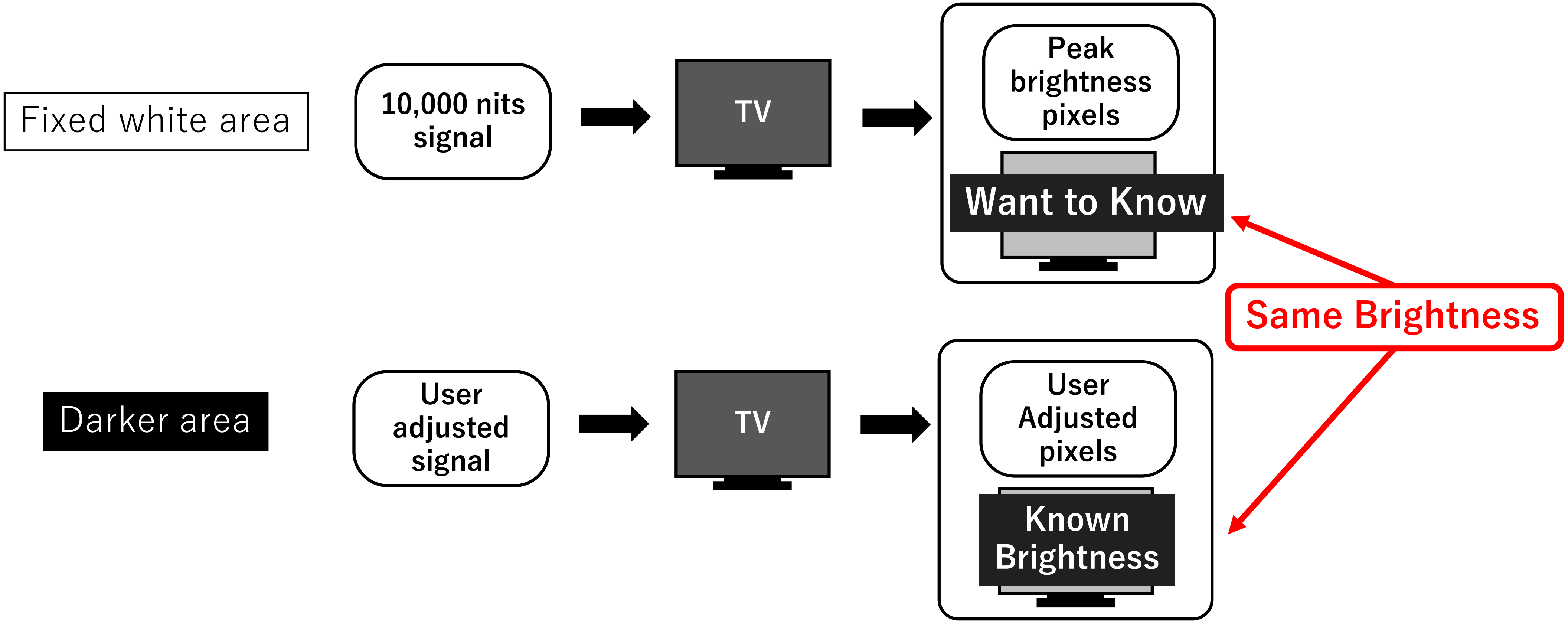
How Can This Estimate the Peak Brightness?

- When the user adjusted output is indistinguishable from the output of the fixed white area



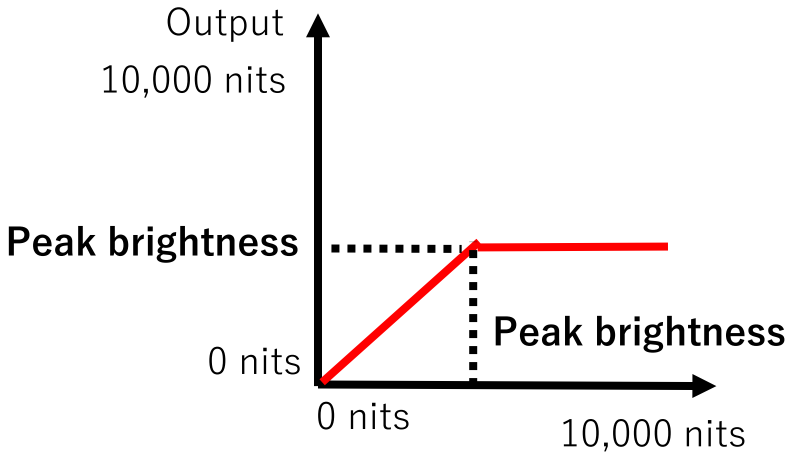
How Can This Estimate the Peak Brightness?

- When the user adjusted output is indistinguishable from the output of the fixed white area



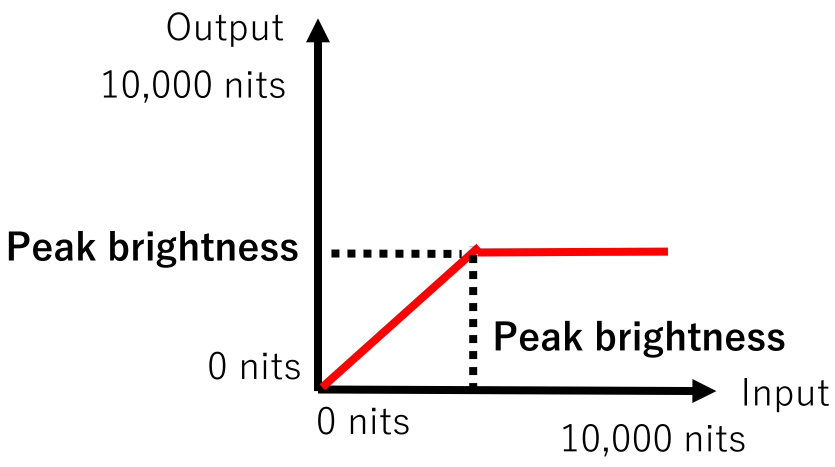
Important Points (1)

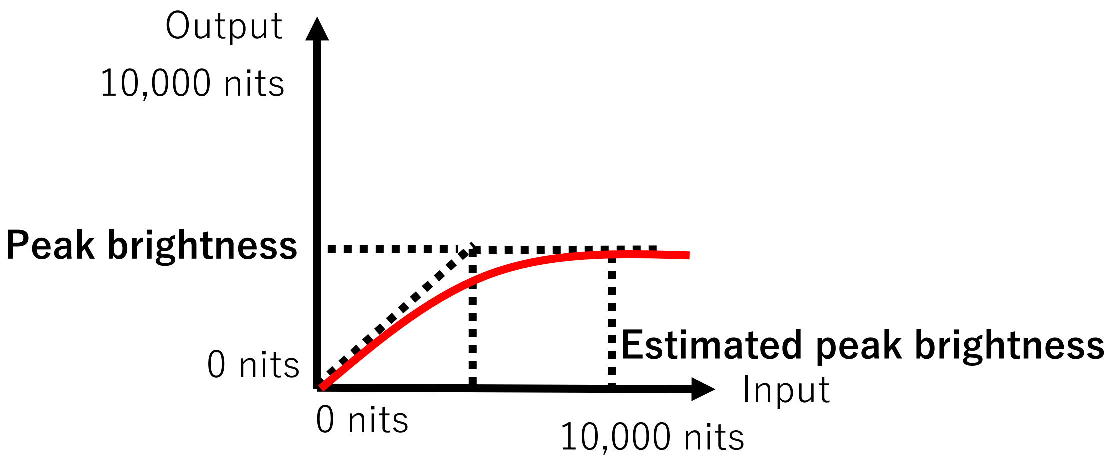
- This process assumes that devices directly map the input signal to the output brightness (the left figure)



Important Points (1)

- This process assumes that devices directly map the input signal to the output brightness (the left figure)
- Actual behavior is often the right figure





Overestimation

Important Points (2)

- This process also assumes that the peak brightness is always constant on the target device

The same input signal, the same output brightness

Important Points (2)

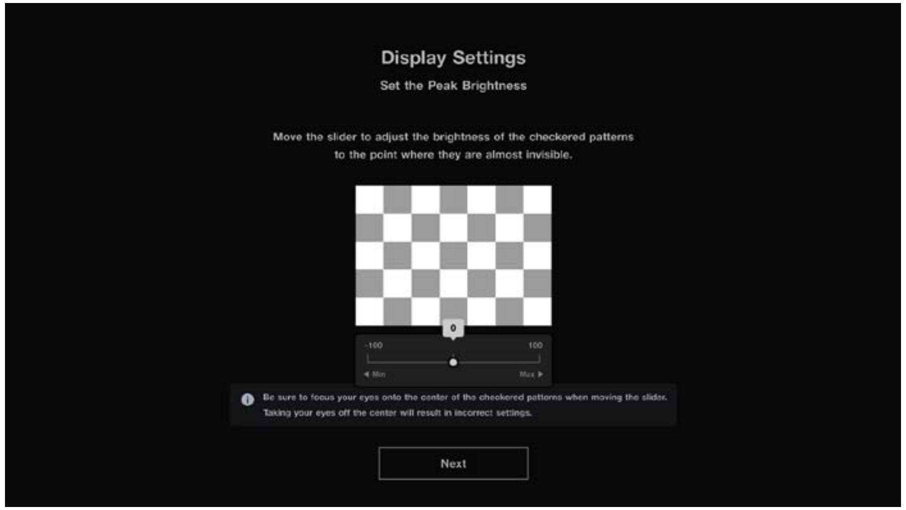
- This process also assumes that the peak brightness is always constant on the target device
- Actual behavior depends on the entire output image
  - This happens because of each device's power management features[HGIG 2018]

The same input signal, **but the output can be different**

Animation

Important Points (2)

- The peak brightness of the full screen is not necessary to estimate
  - It is not common to present a peak-brightness picture on the full screen
- This is why we use the checker board pattern on a part of the screen and not full screen



Results

- Comparison between measured brightness and estimated value

	Measured Brightness (nits)	Estimated Value (nits)
Mid-end TV	400	1100
High-end TV	1200	2000
Master monitor	410	470
Mid-end OLED TV	700	4300

This was measured using a luminance meter, but these values are not very accurate.
These values depends on devices and the measurement settings and are here as a reference

1. Introduction of This Course (5 min.)
2. Introduction of Color and HDR (20 min.)
3. Practical HDR Output Techniques (15 min.)
4. Asset Creation (15 min.)
5. Implementations and Results in Our Game (30 min.)
    a. Implementation of GT Tone Mapping for HDR Output
    b. Tone Mapping Calibration for Output Devices
    c. Rendering in a Wide Color Space and HDR
    d. HDR Color Grading
6. HDR Working Environment (10 min.)

Rendering in a Wide Color Space and HDR

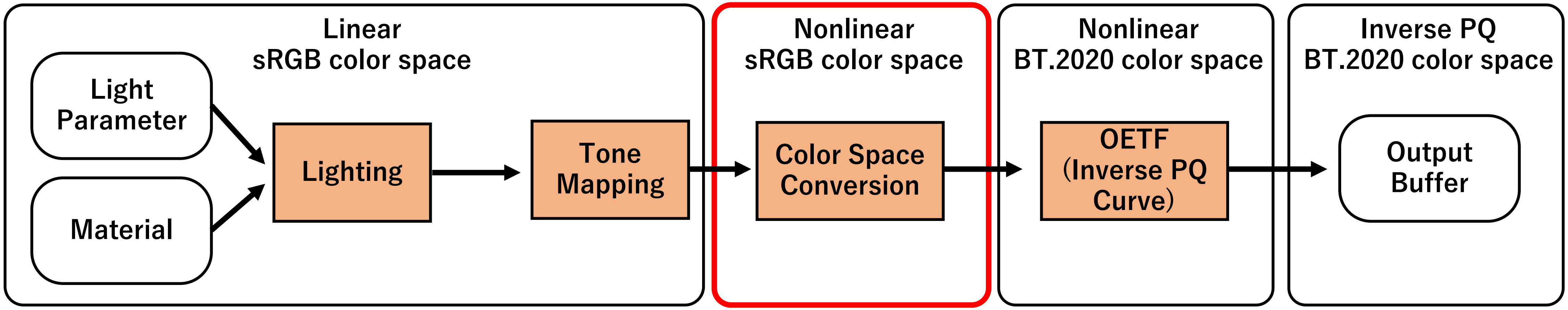
- For HDR rendering, we render frames in a wide color space and apply inverse PQ curve as OETF
  - Wide color rendering
  - Inverse PQ curve

“HDR” Rendering

- Recently, “HDR” rendering has become the means to generating making output for HDR ready devices
  - In the past, we used to render on a linear space and a high dynamic range

Conservative Approach for HDR Rendering

- Rendering in a narrow, conventional sRGB color space, then convert them for the wide, BT.2020 color space to output
  - Every sRGB color can be represented in the BT.2020
  - Conversion is very simple and easy



Conservative Approach for HDR Rendering

- In this approach, the rendering process itself doesn’t need to be changed a lot
  - No need to convert assets
  - Other assets may be easily expanded to new color space
    - E.g. particles, light sources

Conservative Approach for HDR Rendering

- In this approach, the rendering process itself doesn't need to be changed a lot
  - No need to convert assets
  - Other assets may be easily expanded to new color space
    - E.g. particles, light sources
- **But, this approach can't make full use of a wide color and HDR output**
  - Limited color space assets
  - Limited rendering color space

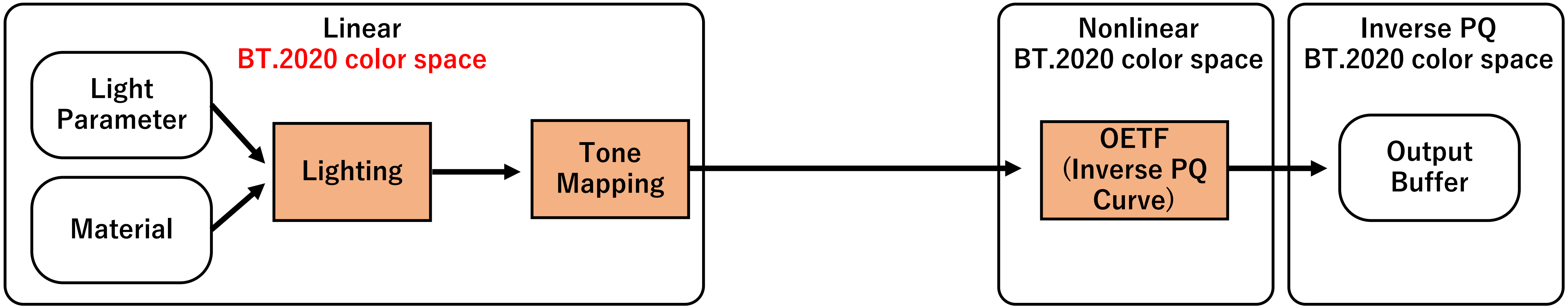
Wide color materials are very common

Wide color materials are very common

Wide color materials are very common

Our Approach for HDR Rendering

- Our approach for outputting to HDR devices is:
  - Rendering scenes in a high dynamic range, linear space
  - **Rendering scenes in a wide color space (BT.2020 in HDR10)**
    - Frame buffer is R11G11B10 (32bits/pixel) to reduce bandwidth



Our Approach for HDR Rendering

- Rendering color space candidates [Uchimura 2017]
  - scRGB
  - CIE XYZ
  - ACEScg (ACES computer graphics space)
  - BT.2020

scRGB

- Similar to sRGB color space and uses negative values
- **Pros**:
  - Good compatibility with sRGB (basically the same value)
- **Cons**:
  - Our framebuffers can't store negative values

CIE XYZ

- Basic building block of color science
- **Pros**:
  - Very wide color space
- **Cons**:
  - Too wide and causes large error during lighting or asset conversion
  - White point specification differs from BT.2020 (HDR10)
    - Color shift can occur easily

ACEScg

- Recently popular
- **Pros**:
  - Standard color transformation is well defined
    - RRT (Reference Rendering Transform)
      ODT (Output Device Transform)
  - Good portability
- **Cons**:
  - Many magic numbers exist in RRT and ODT
  - Some color grading processes are included in ODT
    - But we want a neutral one

BT.2020

- Used in HDR10
- **Pros**:
  - No need to convert for HDR10 output
  - The same white point specification as sRGB
    - Color space conversion between sRGB and BT.2020 is stable (no color shift)

- Looked good however…
  - No previous work using BT.2020 for rendering color space at that time

BT.2020

- Used in HDR10
- **Pros**:
  - No need to convert for HDR10 output
  - The same white point specification as sRGB
    - Color space conversion between sRGB and BT.2020 is stable (no color shift)

- Looked good however…
  - No previous work using BT.2020 for rendering color space at that time

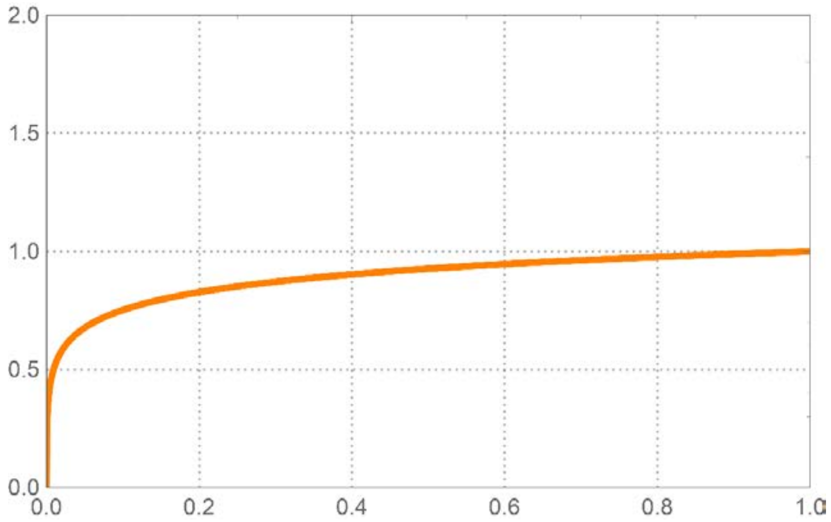
→ BT.2020 was the best solution at the time compared to other solutions, so we decided to use it for the rendering color space.

Inverse PQ Curve

- After rendering (in BT.2020), we need to apply the inverse PQ curve for HDR10 output
- Inverse PQ curve is described below (again)

```
Float InversePQ(float3 input_RGB)
{
  const float m1 = 0.1593017578125;
  const float m2 = 78.84375;
  const float c1 = 0.8359375;
  const float c2 = 18.8515625;
  const float c3 = 18.6875;

  float3 ym = pow(input_RGB, m1);
  return pow((c1 + c2 * ym)/(1 + c3 * ym), m2);
}
```



[0, 10000] nits is mapped into [0, 1] for the function input

Inverse PQ Curve

- Need a fast and accurate approximation of inverse PQ
  - Why fast?
    - Inverse PQ is applied every frame, so the performance is important
  - Why accurate?
    - To isolate problems easily and properly in a whole workflow

Approximations of Inverse PQ

- Some approximations are proposed
  - [Patry 2017]
    - Fast analytic approximation
    - Valid under about 4,000 nits and not so accurate
  - [Malin 2018]
    - Lookup based approach
    - Looks fast
    - May be not accurate enough
    - Didn’t know about it during development

Approximations of Inverse PQ

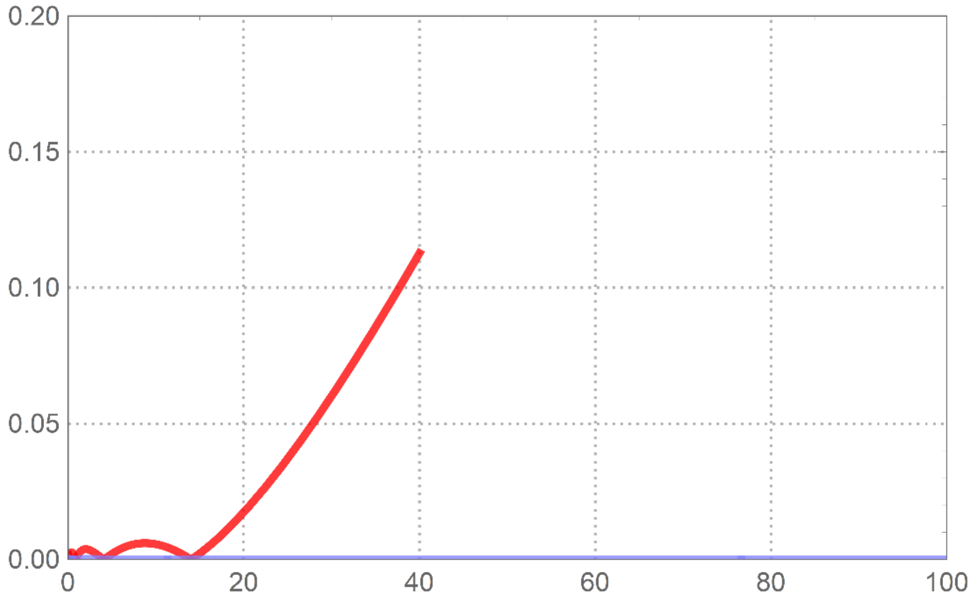
- Our approximation
  - Fitted by Mathematica

```
float InversePQ_approx(float x)
{
    float k = pow((x * 0.01f), 0.1593017578125);
    return
        (3.61972*(1e-8) + k * (0.00102859 + k * (-0.101284 + 2.05784 * k))) /
        (0.0495245 + k * (0.135214 + k * (0.772669 + k)));
}
```

In advance, [0, 10000] nits is mapped into [0, 100] for the function input

Approximations of Inverse PQ

- Comparison of the accuracy to [Patry 2017]
  - Desmos graph: https://www.desmos.com/calculator/0n402k2syc
    - Absolute error to the original inverse PQ
  - The bright input part of our approximation is much more accurate



Purple line: our approximation
Red line: [Patry 2017]

Performance Results

- PlayStation®4, 1920x1080 (includes tone mapping)

Reference	0.36 ms
Our Approximation	**0.33 ms (10% faster than reference)**
[Patry 2017]	0.29 ms

1. Introduction of This Course (5 min.)
2. Introduction of Color and HDR (20 min.)
3. Practical HDR Output Techniques (15 min.)
4. Asset Creation (10 min.)
5. Implementations and Results in Our Game (30 min.)
    a. Implementation of GT Tone Mapping for HDR Output
    b. Tone Mapping Calibration for Output Devices
    c. Rendering in a Wide Color Space and HDR
    d. HDR Color Grading
6. HDR Working Environment (10 min.)

Color Grading

- Color grading is the process of modifying input images in various ways to enhance final look
  - Brightness
  - Contrast
  - Hue
  - Etc

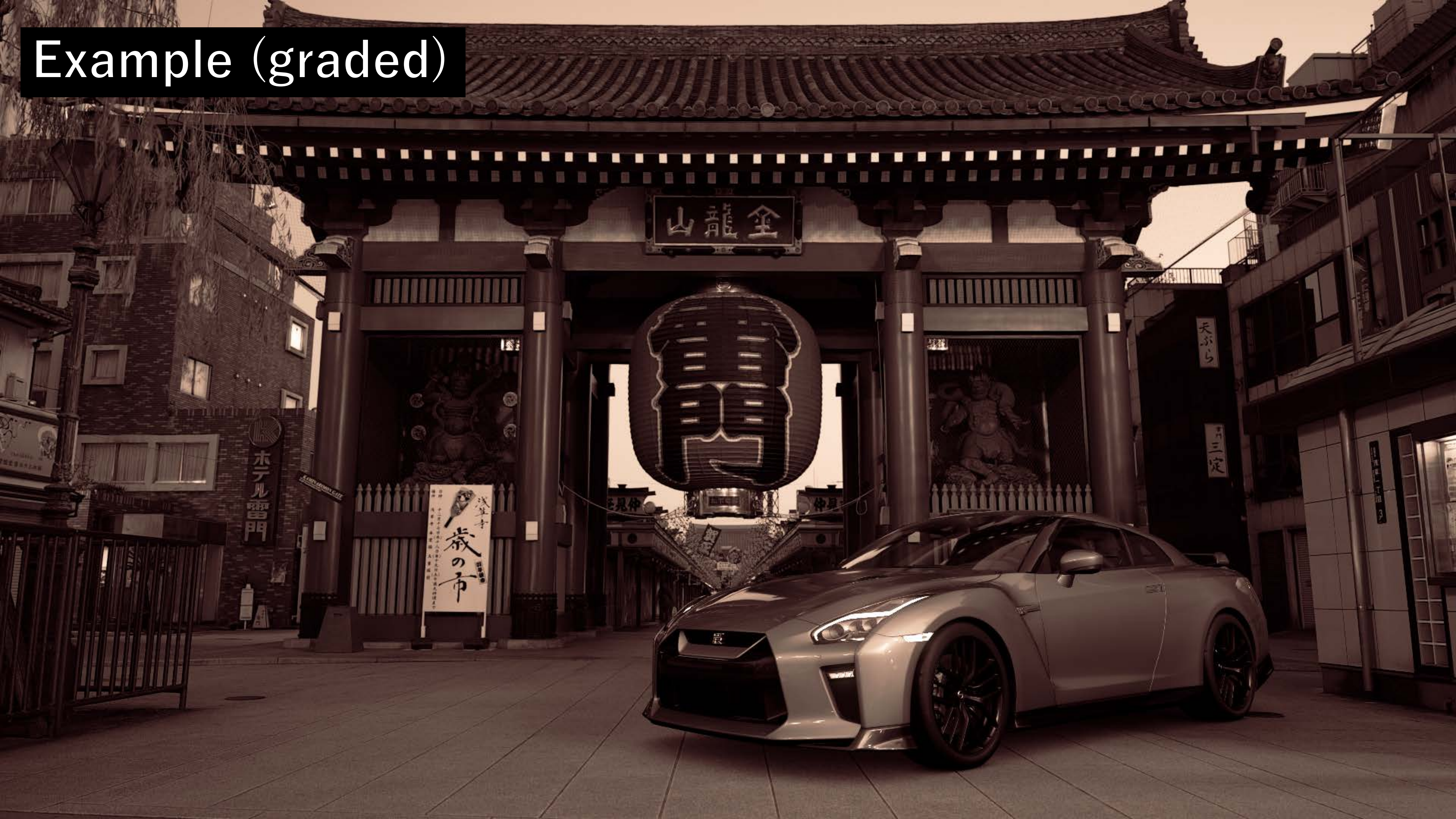
- Originally from the movie industry
  - In the past: directly applied to the film itself
    - E.g. bleach bypass
  - Today: digital processing workflow

Color Grading in Gran Turismo SPORT

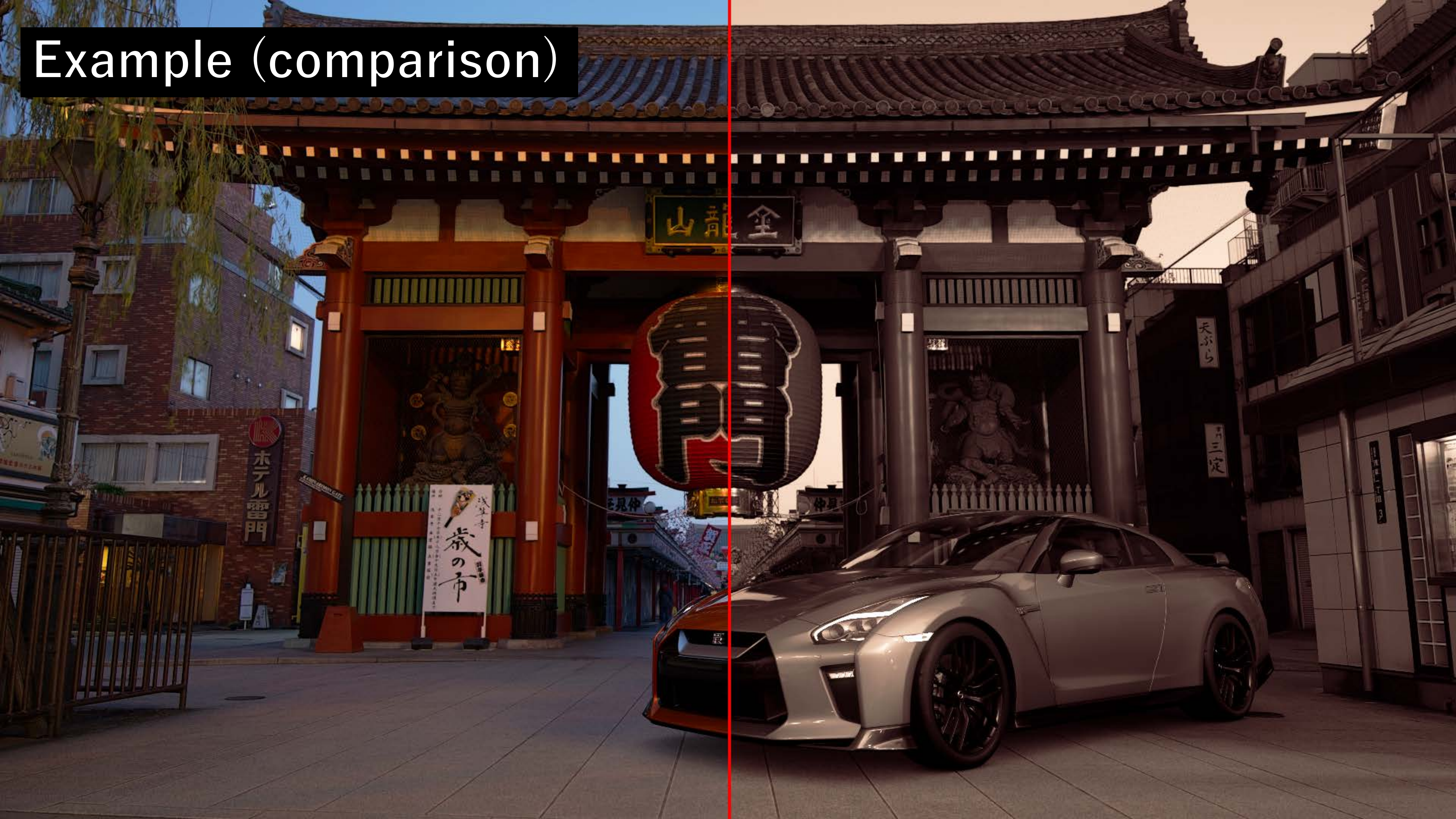
- Artists can enhance appearances in a cut scene
  - For attracting users with impressive images
- Users can edit in-game taken photos by in-game color grading tools
  - Race photo mode
  - Scapes photo mode

Example (original)

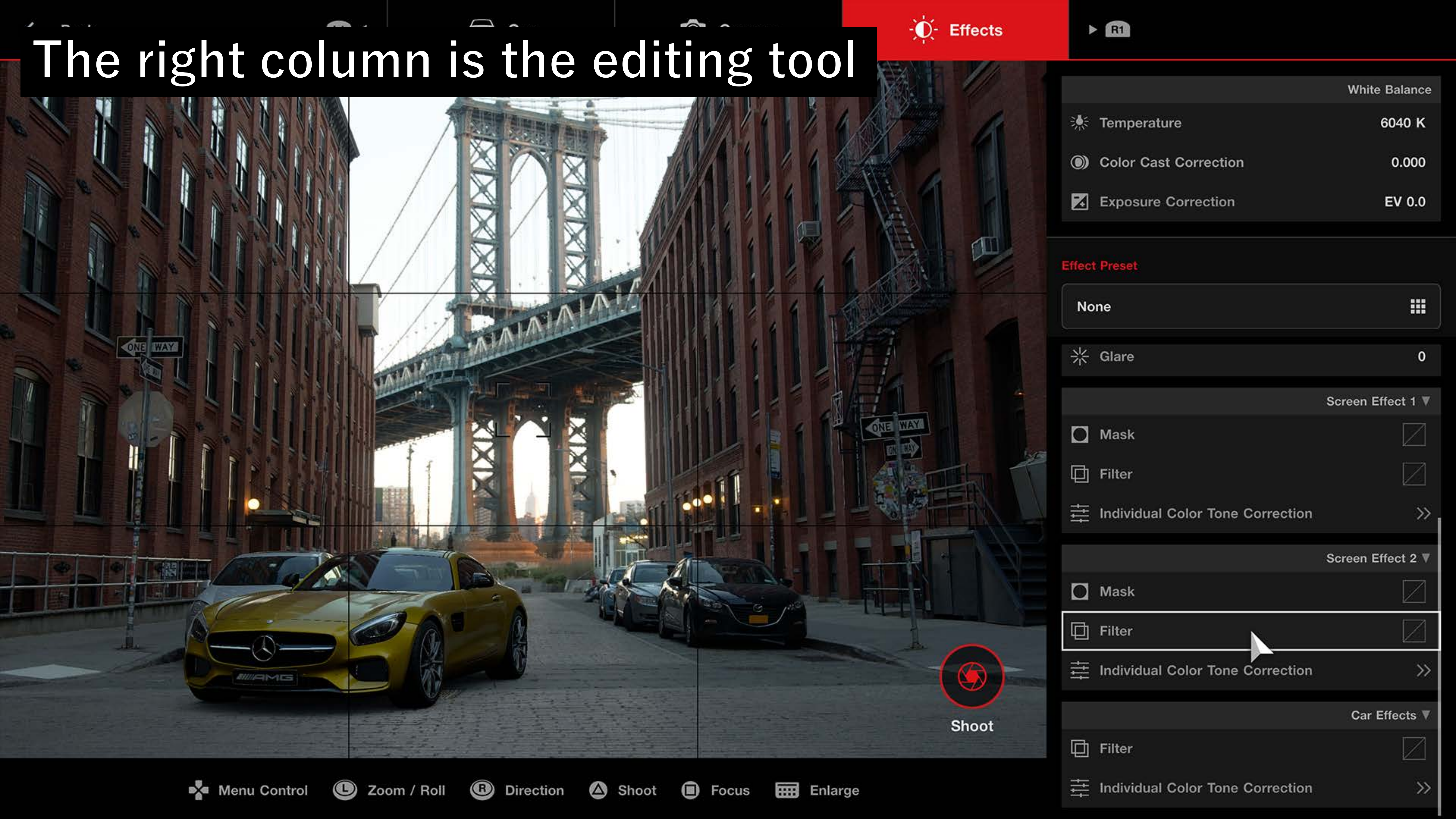
Example (graded)



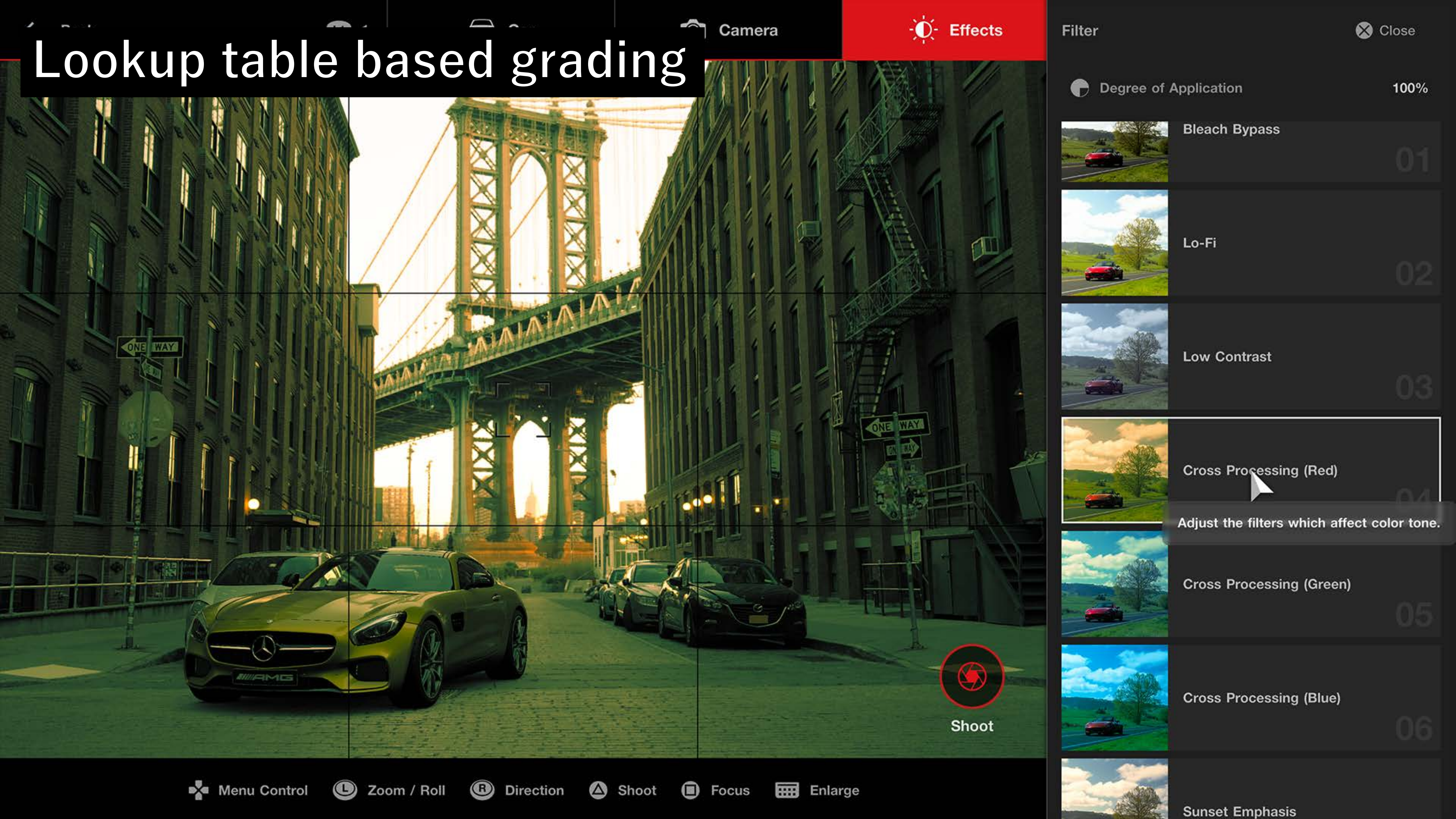
Example (comparison)

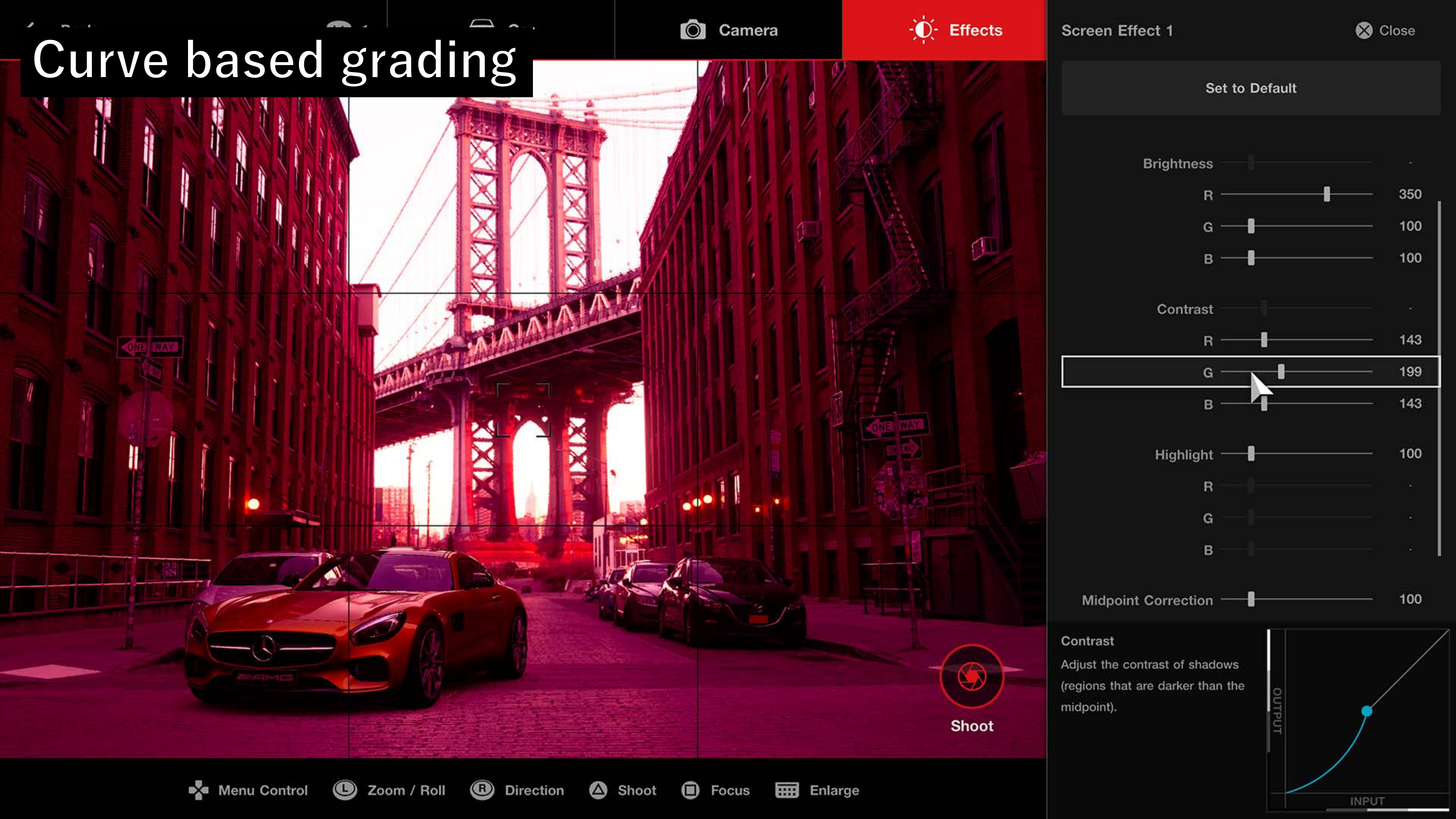


The right column is the editing tool



Lookup table based grading




Curve based grading
Camera
Effects
Screen Effect 1
Close
Set to Default
Brightness
R 350
G 100
B 100
Contrast
R 143
G 199
B 143
Highlight 100
R
G
B
Midpoint Correction 100
Contrast
Adjust the contrast of shadows (regions that are darker than the midpoint).
OUTPUT
INPUT
Shoot
Menu Control
Zoom / Roll
Direction
Shoot
Focus
Enlarge
ONE WAY
ONE WAY

Color Grading for HDR Output

- Color grading is important
  - In SDR, for “more” artistic looking

➡ We also want to apply color grading in HDR

Our Goal

- **Support both lookup table (LUT) based grading and parametric curve based grading**
  - Lookup table based approach: commonly used but hard to create
  - Parametric curve based approach: limited but easy to use

- **Compatible solution between SDR and HDR**
  - Want to reduce grading cost

Lookup Table Based Grading (SDR)

- In SDR, LUT based grading is commonly used
  - Can be graded by any grading software
    - E.g. DaVinci Resolve, Adobe Photo Shop

- Why LUT?
  - To reproduce complex grading process easily

→ Baking a grading process into 3D lookup tables ☺

Lookup Table Based Grading (SDR)

- Using 3D LUT, arbitrary grading functions can be approximated in a piecewise-linear manner

$$\left(R_{graded}, G_{graded}, B_{graded}\right) = f_{grading}(R, G, B)$$

⬇

```
float3 graded_color = LUT3D.Sample(trilinearSampler, input_color).rgb;
```

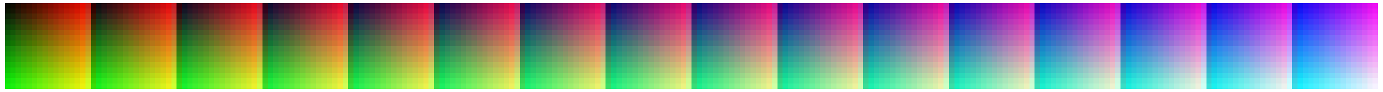
Computing original function can be approximated by single 3D texture fetching

Lookup Table Based Grading (SDR)

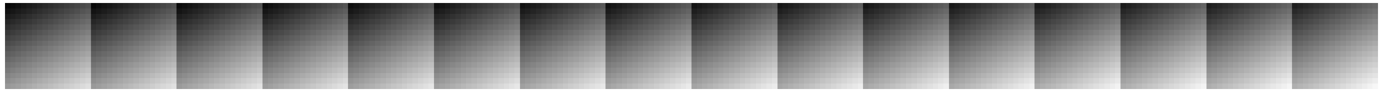
- SDR 3D LUT is simple and efficient

Resolutions	16x16x16 is enough
Texture bit-depth	8 bit and is compact ☺

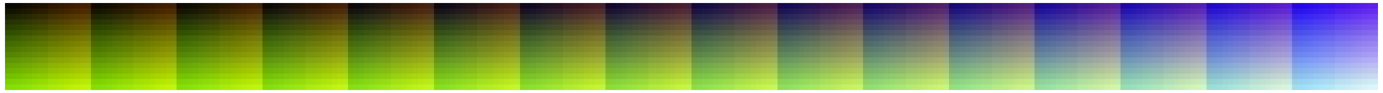
- Examples



Neutral LUT (No-operation)



Monotone LUT



Something complex

Lookup Table Based Grading (HDR)

- The main difference is an input and output format

	Color Space	OETF
SDR	sRGB	sRGB gamma
HDR	BT.2020	Inverse PQ curve

- Applying process is the same, but grading and creation process is different
  - Need to do color grading and create 3D LUT independently for HDR
  - Independently for each HDR device (each peak brightness)

Lookup Table Based Grading (HDR)

- Need to prepare many color grading settings ☹
  - For both SDR and HDR
  - For each HDR device

➡ High Cost!

Lookup Table Based Grading (HDR)

- Need to prepare many color grading settings ☹
  - For both SDR and HDR
  - For each HDR device

➡ High Cost!

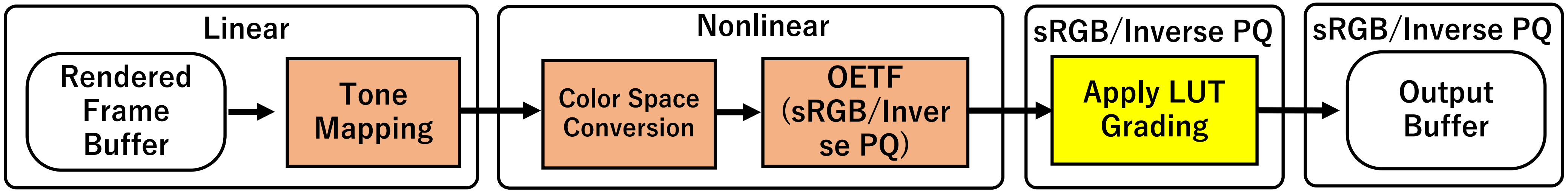
➡ Want to use single LUT for SDR and HDR (all device targets), so we developed a new compatible solution.

SDR/HDR Compatible LUT Grading

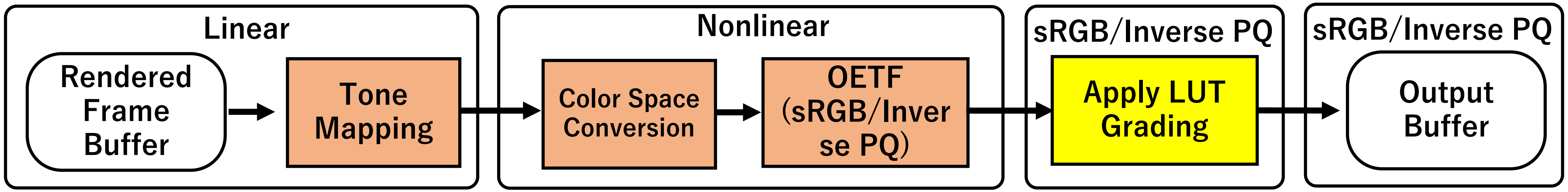
- Once color grading is done (and make LUT), use it everywhere
  - The grading target is set for the most generic environment

Color Space	BT.2020
OETF	Inverse PQ curve
Target nits	10,000 nits

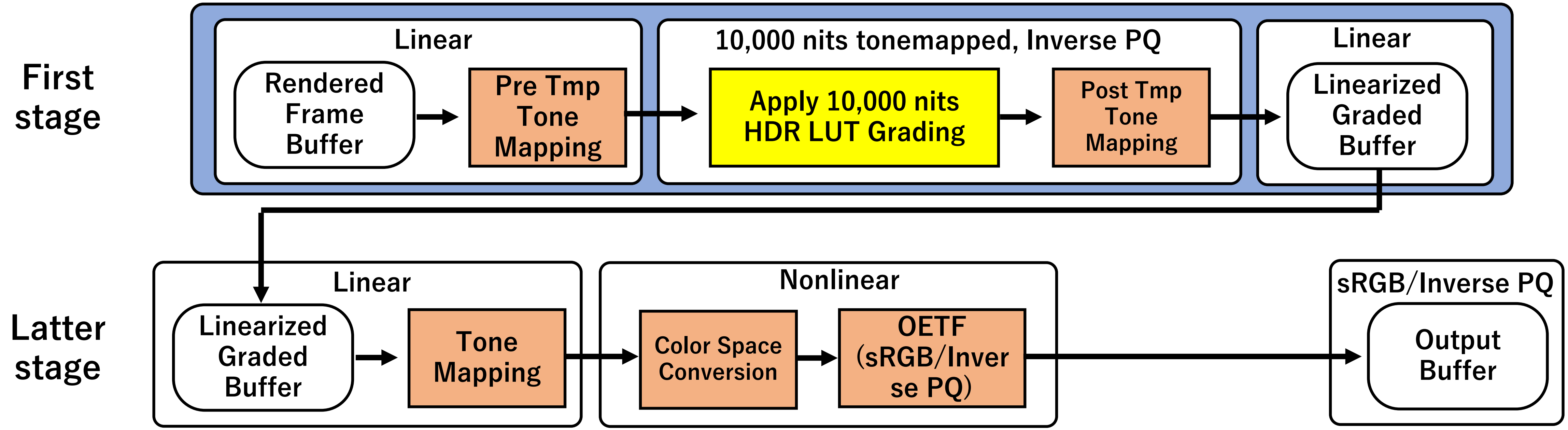
- Software
  - DaVinci Resolve
    - de facto standard in the movie and game industry
  - For preview purposes, apply an appropriate display LUT



Ordinary Color Grading Process



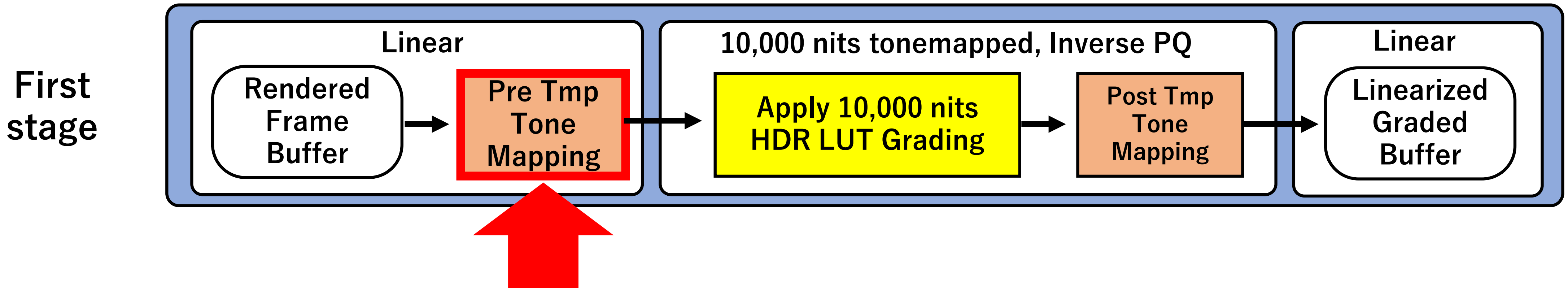
Ordinary Color Grading Process



Our SDR/HDR Compatible Grading Process

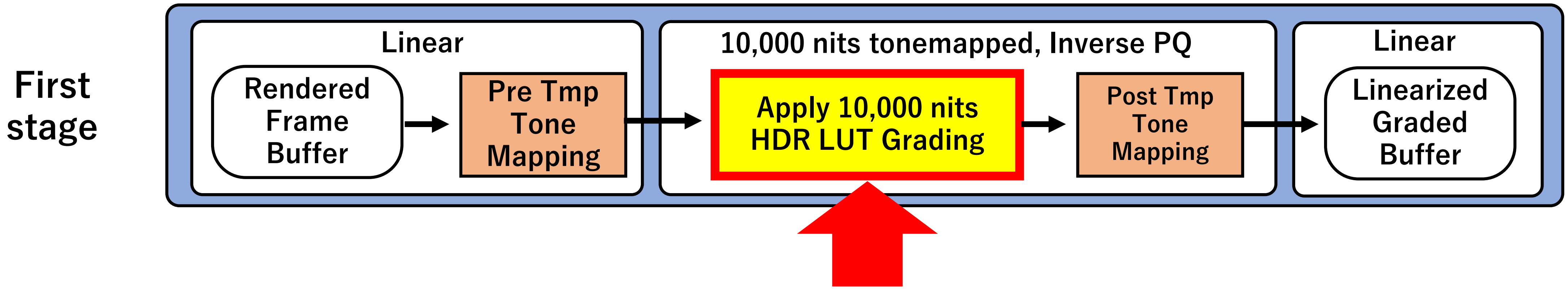
Our SDR/HDR Compatible Grading Process

1. Pre temporal tone mapping is applied to a linear rendered buffer
   - Apply GT Tone Mapping (target 10,000nits) and then inverse PQ curve
   - The buffer is completely valid for 10,000 nits target HDR10 format



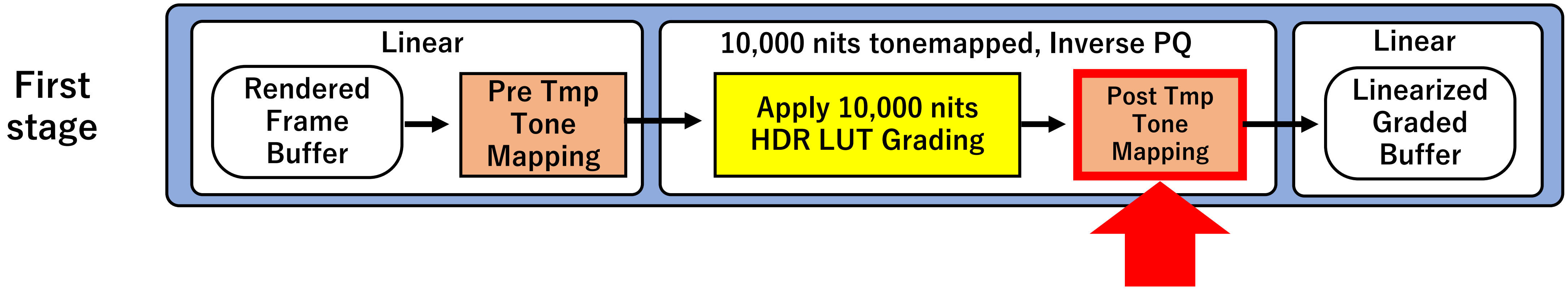
Our SDR/HDR Compatible Grading Process

2. HDR 3D LUT (target 10,000 nits) is applied in an ordinary way
   - This process is valid because the input buffer’s target and color space are the same as grading lookup table



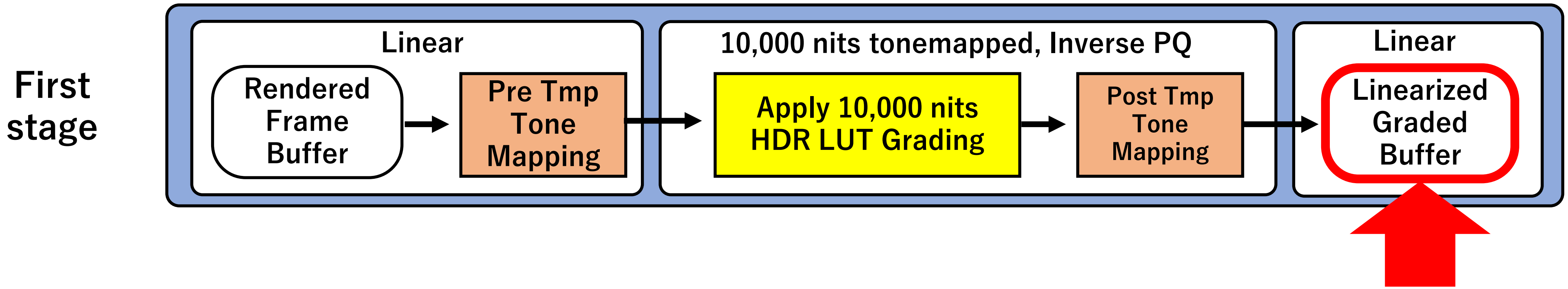
Our SDR/HDR Compatible Grading Process

3. Post temporal tone mapping is applied
   - Apply PQ curve (not inverse PQ) and then inverse GT Tone Mapping to reconstruct a linear buffer



Our SDR/HDR Compatible Grading Process

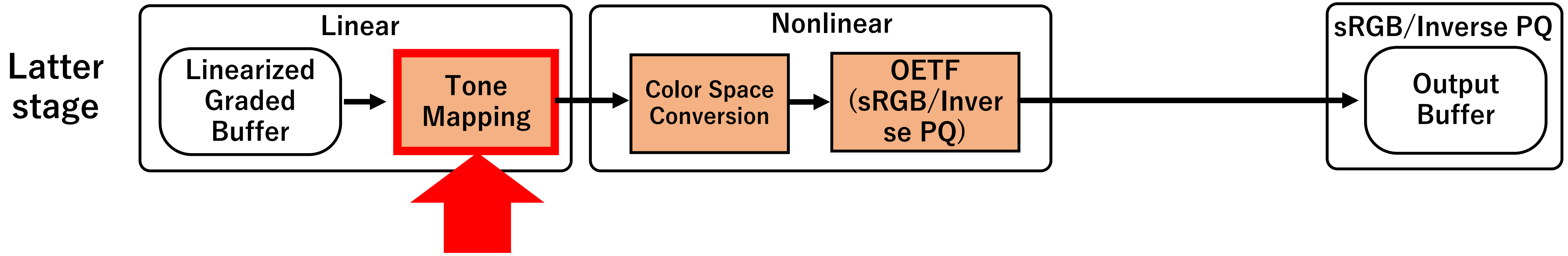
4. Finally, we can gain the linearized, color graded buffer



SDR/HDR Compatible LUT Grading

- After grading
  - Apply GT Tone Mapping to the linear, graded buffer

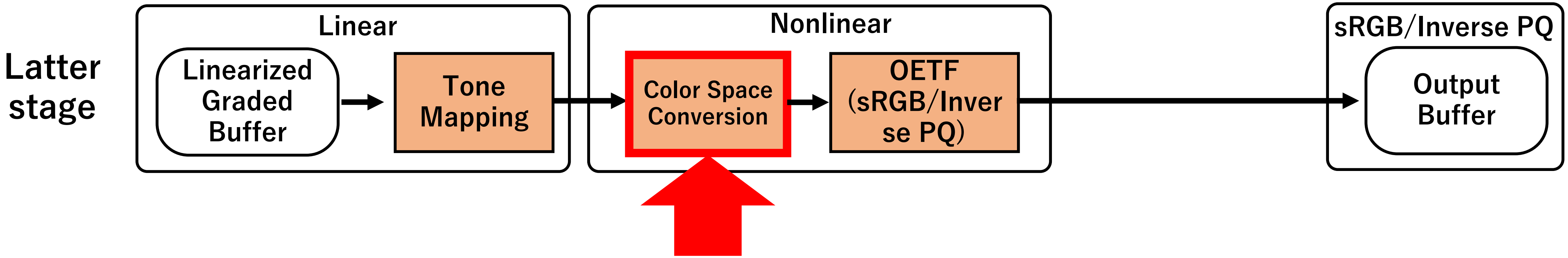
SDR	SDR tone mapping
HDR	HDR tone mapping for each output device



SDR/HDR Compatible LUT Grading

- After grading
  - Apply the color space conversion

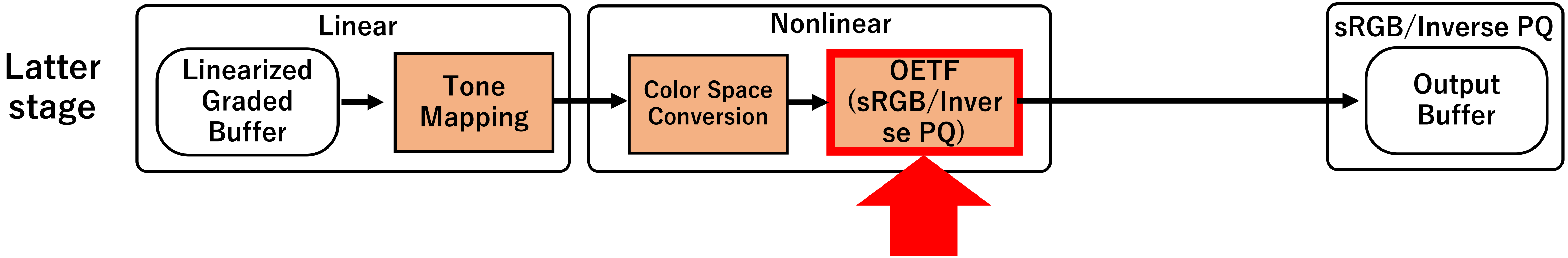
SDR	BT.2020 color is converted to sRGB color
HDR	Nothing happens (the same color space in HDR10)



SDR/HDR Compatible LUT Grading

- After grading
  - Apply OETF

SDR	sRGB gamma
HDR	Inverse PQ curve



Results

SDR (LUT off)

HDR (LUT off)

These are the same scene
Directly captured from TV output by a still camera

SDR (LUT off)

HDR (LUT off)

In HDR, the dynamic range is better preserved than SDR
(In SDR, wider area is saturated because of its narrow range)

Results

SDR (LUT on)

HDR (LUT on)

The same LUT is applied in both SDR and HDR
The whole impression looks similar ☺

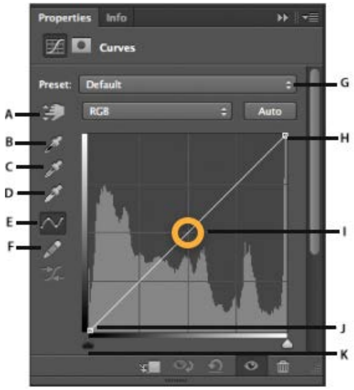
Results

SDR (LUT off)

HDR (LUT off)

Parametric Curve Based Grading

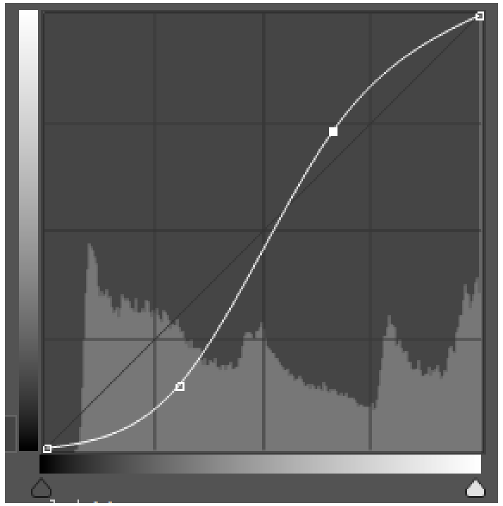
- LUT is great but often hard to create
  - Many methods to adjust
  - Not easy to learn grading process
- Simple, parametric curve based grading
  - Want a method like the “Curve” tool in Adobe Photoshop, but we need a tool compatible with both SDR and HDR.



https://helpx.adobe.com/jp/photoshop/using/curves-adjustment.html

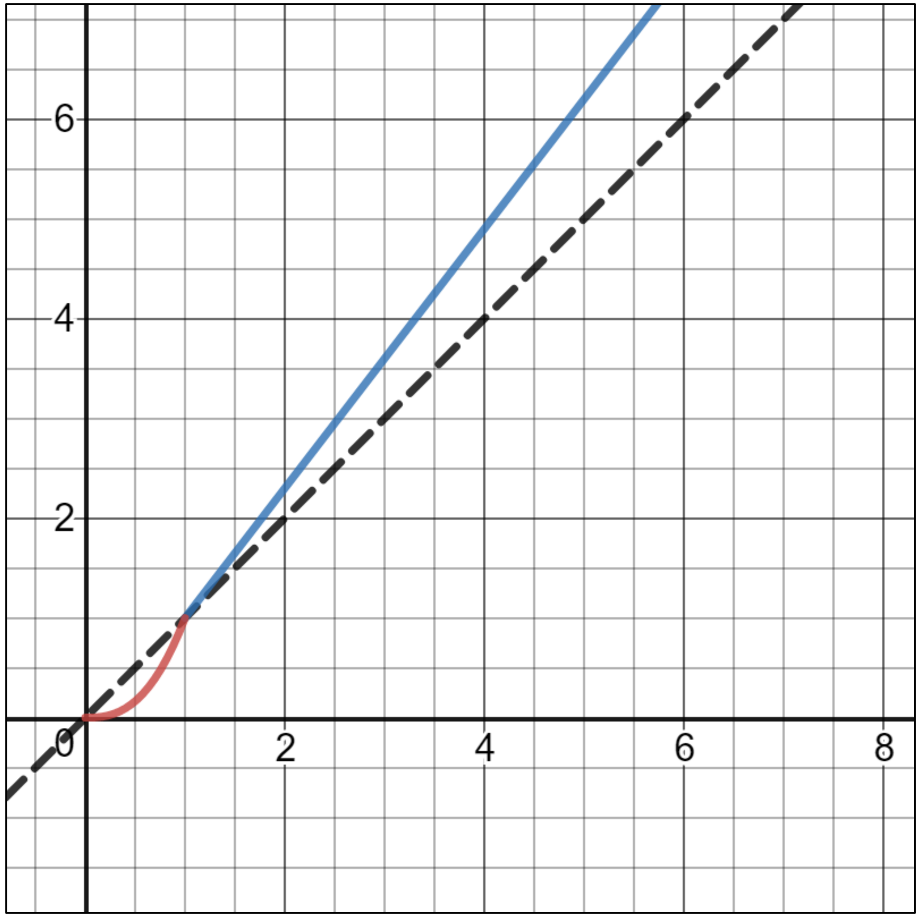
Parametric Curve Based Grading

- Requests from artists
  - Wanted to enhance contrasts easily!
- In SDR, “S” curve [Hable 2010] can be a solution but in HDR?
  - Apply directly in a linear space
  - Only dark part needs to be bended
  - Bright part can’t be bended
    - The range is not [0, 1]
    - Tone mapping will bend



Parametric Curve Based Grading

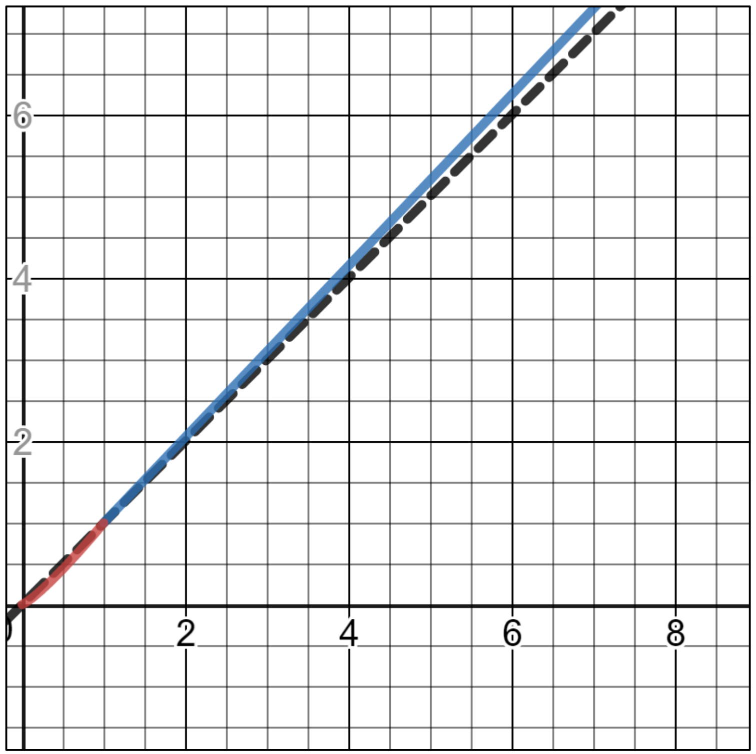
- **Our solution**
  - The dark part is a simple gamma curve
  - The bright part is just a line!
  - 3 parameters
    - DarkPartGamma: p
    - MidPoint: m
    - BrightPartScaling: c



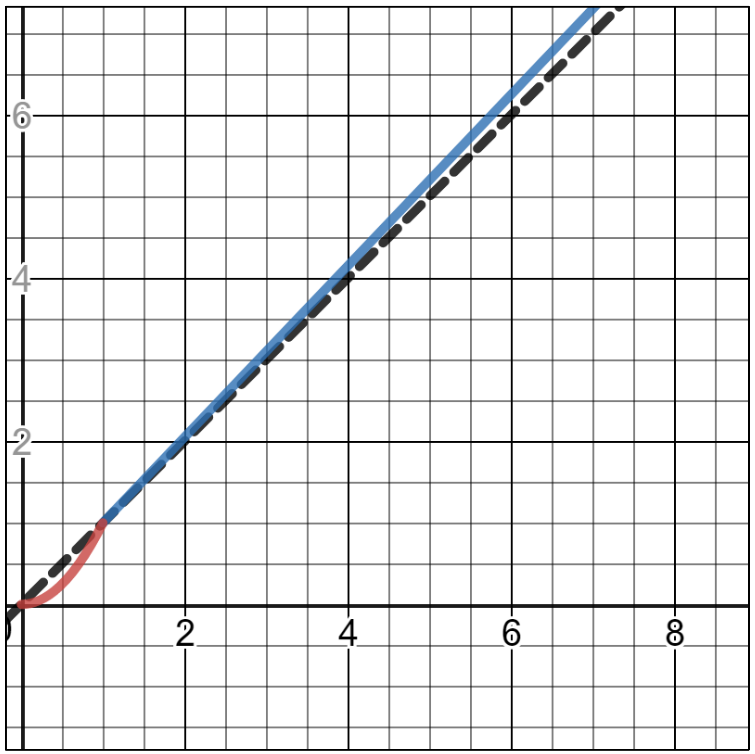
$$f(x) = \begin{cases} \dfrac{x^p}{m^{p-1}}, & 0 < x < m \\ c(x-m)+m, & m \leq x \end{cases}$$

x is an individual component of RGB and f(x) is graded value

Parametric Curve Based Grading

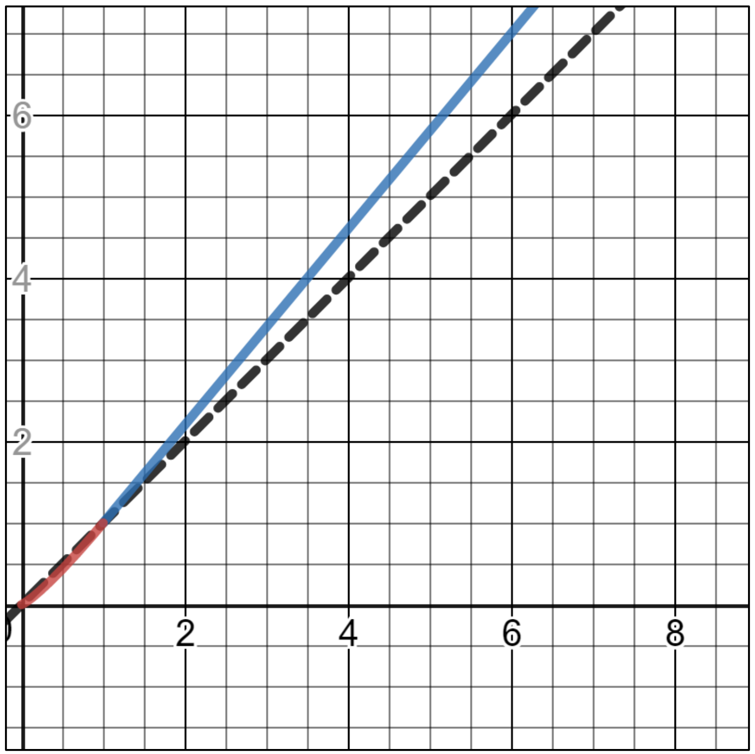


$p = 1.2, m = 1, c = 1.05$



$p = 2, m = 1, c = 1.05$

Strong emphasizing a dark part

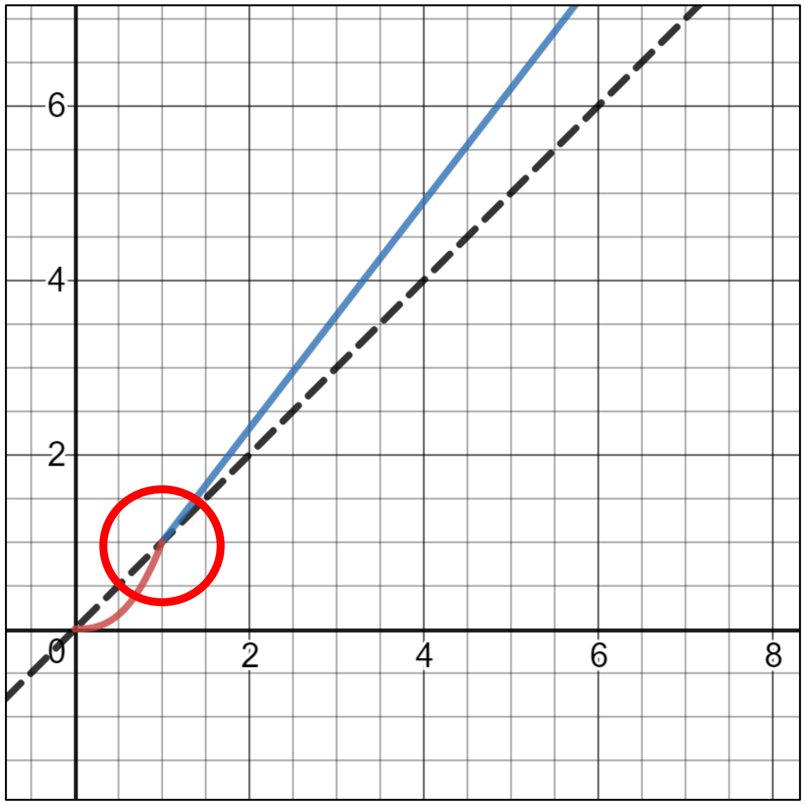


$p = 1.2, m = 1, c = 1.2$

Strong emphasizing a bright part

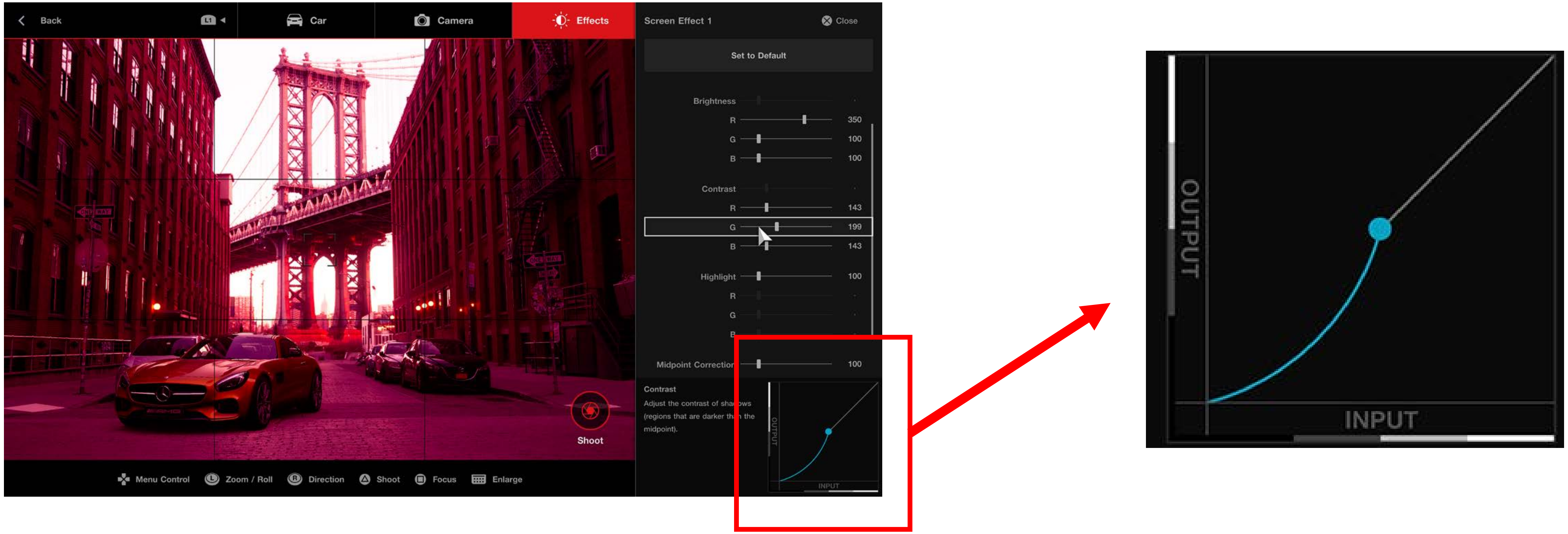
Parametric Curve Based Grading

- The dark part and bright part aren't connected smoothly, but artists liked this curve, so that is OK!



Parametric Curve Based Grading

- The same grading curve is implemented in a photo mode in Gran Turismo SPORT for photo editing tools
- World-wide players are grading their photos in a linear space using a SDR/HDR compatible parametric curve!



Example (original)

Example (graded)

Example (comparison)

Conclusions

- Achieved support of lookup table (LUT) based grading and simple parametric curve based grading
- Both methods are SDR/HDR compatible

Conclusions

- Achieved support of lookup table (LUT) based grading and simple parametric curve based grading
- Both methods are SDR/HDR compatible

→ We achieved the goal: significantly enhance our HDR rendering output.

1. Introduction of This Course (5 min.)
2. Introduction of Color and HDR (25 min.)
3. Practical HDR Output Techniques (10 min.)
4. Asset Creation (15 min.)
5. Implementations and Results in Our Game (30 min.)
6. HDR Working Environment (10 min.)

Build Neutral Image

- Verifying HDR output is very hard.
- Too many variations in HDR TVs.
- Neutral image output is very important.
- Needed for easy debugging.

Watch It Physically

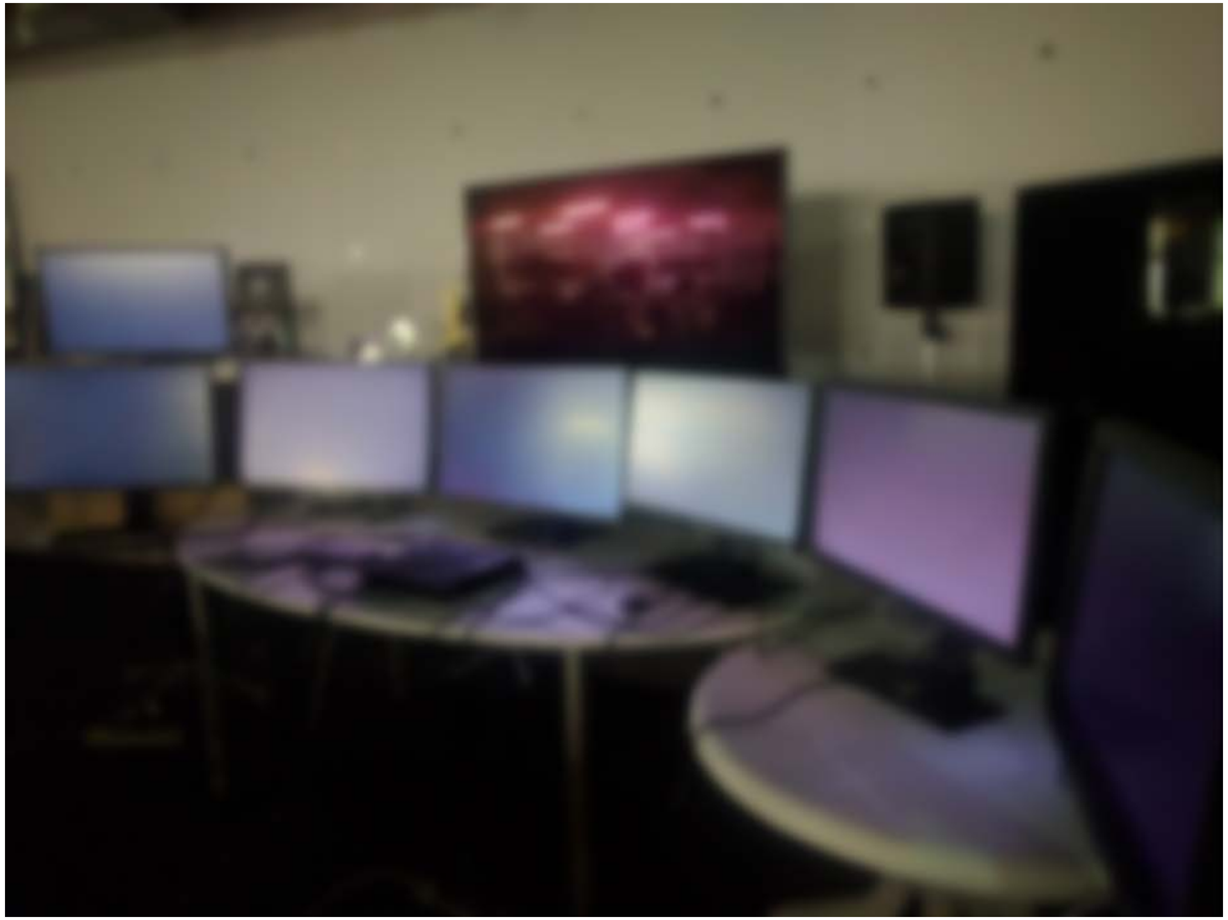
- Specification on internet is not enough to trust.
- Watching and testing actual monitor is important.
- We created our own testing suite.

Test All Monitors

- To test the monitor we established standard testing patterns.
  1. Small window test
  2. Moving small window test
  3. Full locus image
  4. HSV scroller

- and so on.

Some Day at Office

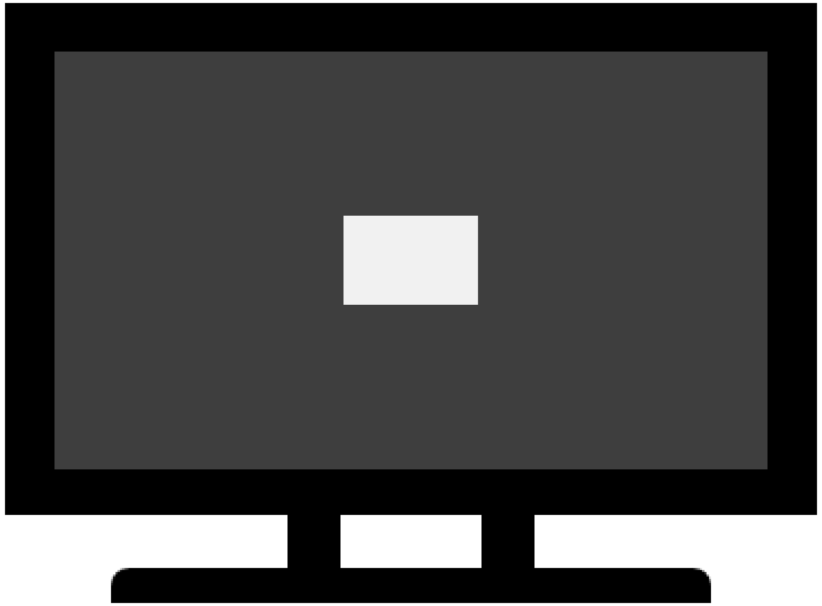


Small Window Test

Small Window Test with Size

Small Window Test

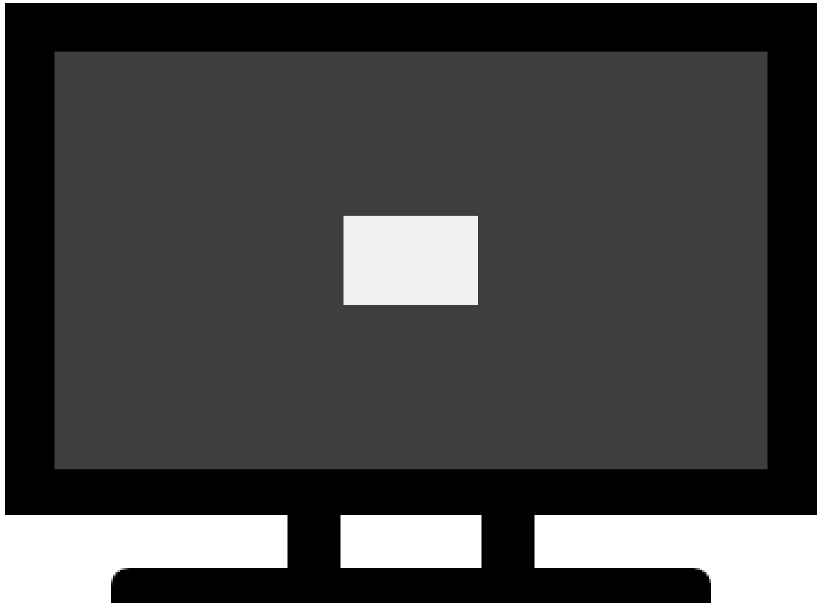
Moving Small Window Test



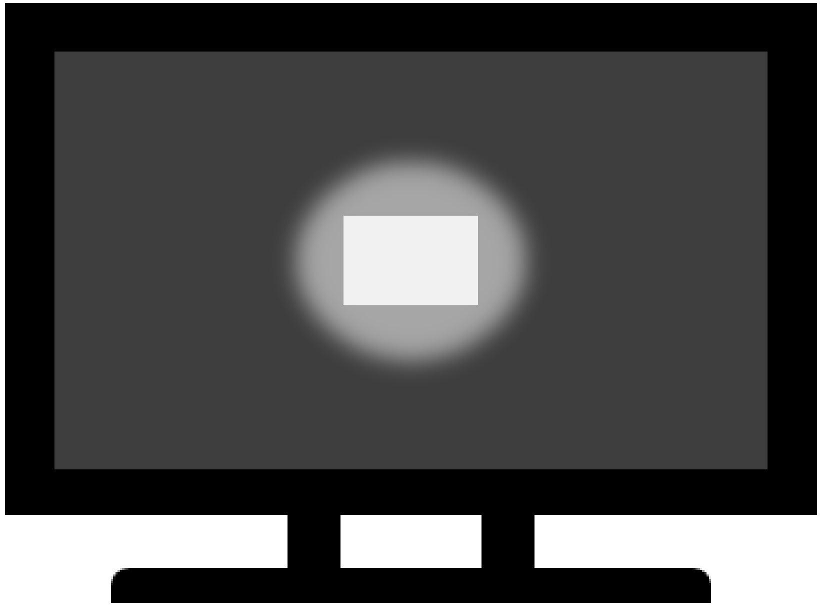
Local Dimming

- Local dimming enhances TV contrast.

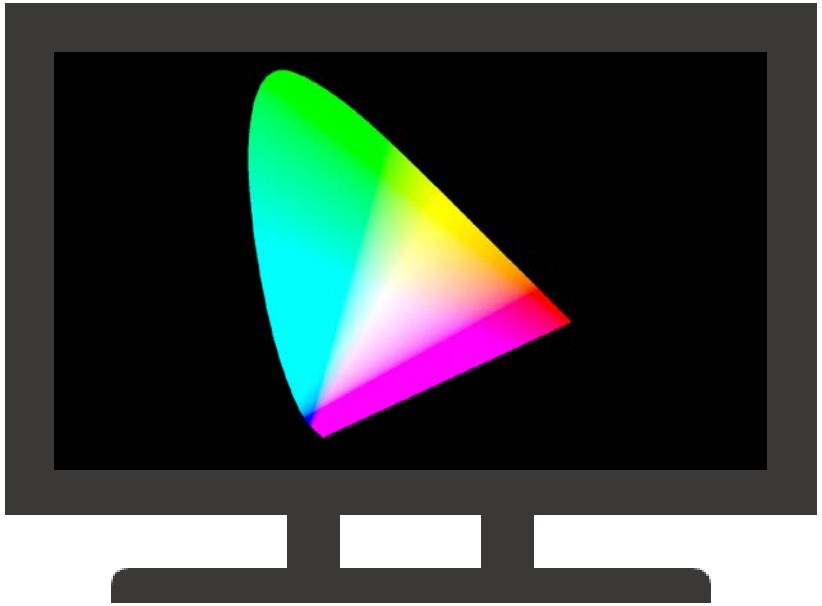
Moving Small Window Test



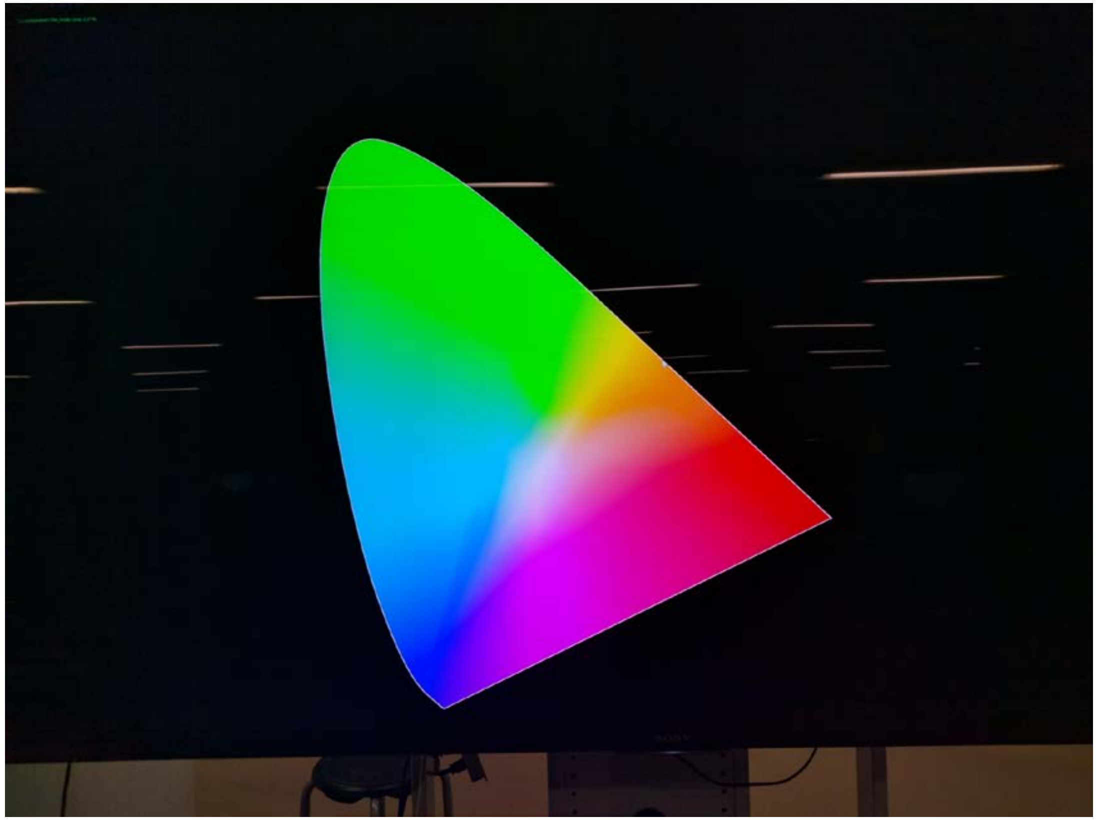
Moving Small Window Test

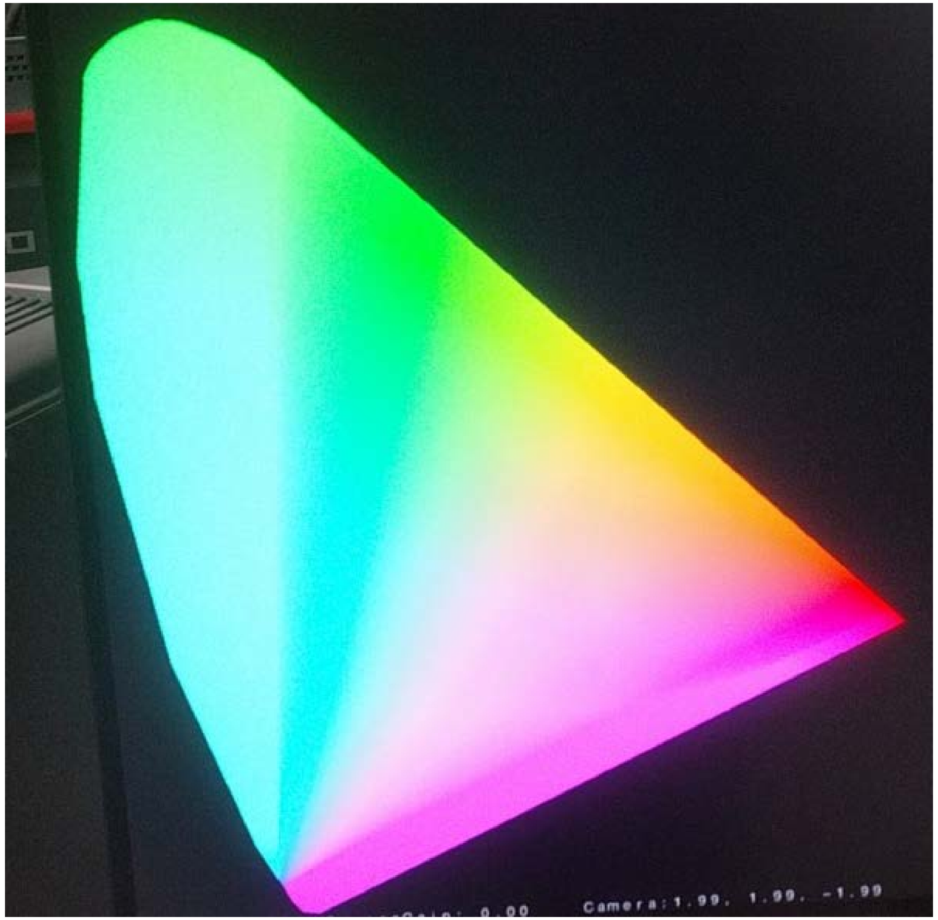


Full Locus Test



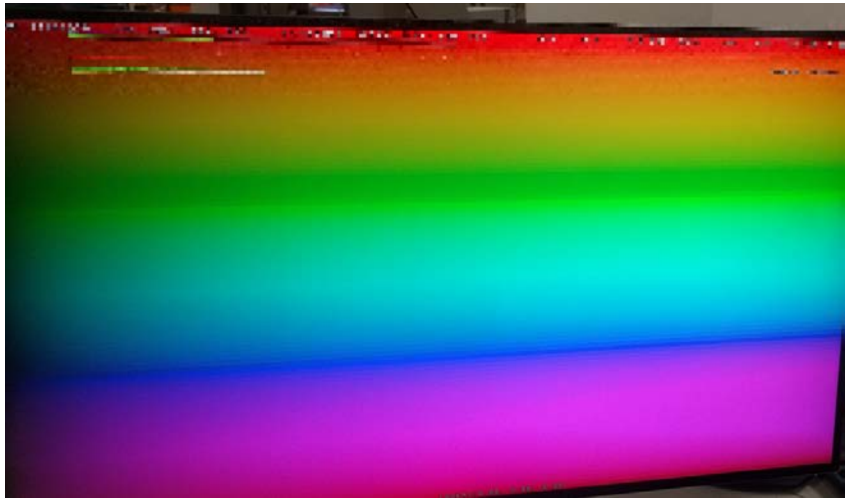
Full Locus Test: Result





HSV Scroller

HSV Scroller Result Example



Conclusion

- HDR world is getting complicated by image enhancement, device limitations, etc..
- To build your believable environment, test everything.

Future Work

- More robust GT Tone Mapping
  - We’d like to have much wider verifications.
  - More precise tone mapping calibration process
- Full spectra rendering
- More effective implementations
  - There are still lot of space to optimize.
- More effective asset capturing.

Wrap up

- We established a consistent theory-based approach for each aspect of the workflow. As a result high quality output is achieved.
- We have talked about fundamental theory and practical implementation about HDR and Wide Color.
  - All of this based on our case study

Wrap up

HDR and Wide Color is good!

Enjoy HDR and Wide Color!


WE ARE HIRING
http://www.polyphony.co.jp/recruit/english/

Acknowledgements

- Development Support
  - Kazunori YAMAUCHI
  - Kenichiro YASUTOMI
  - Shuichi TAKANO
  - Noriyoshi SASAKI
  - SONY Digital Imaging
  - SONY Sound Visual Products
  - SIE HDR Team
- Special Thanks
  - Takuya IZUMI
  - Ren YASUDA
  - Yuki OZAWA
  - Ferreira Charles

References

- [BT1886] ITU-R Rec. BT. 1886. https://www.itu.int/dms pubrec/itu-r/rec/bt/R-REC-BT.1886-0-201103-I!!PDF-E.pdf
- [BT2100] ITU-R Rec. BT. 2100. https://www.itu.int/rec/R-REC-BT.2100-2-201807-I/en
- [CIE 1931] CIE. 1931. Commission Internationale de Iclairage Proceedings. Cambridge: Cambridge University Press.
- [Brindley 1955] G.S. Brindley. 1955. The colour of light of very long wavelength. The Journal of Physiology 130 (1955), 35–44.
- [Chan 2015] Danny Chan. 2015. Real-World Measurements for Call of Duty: Advanced War-fare. (2015). https://research.activision.com/t5/Publications/Real-World-Measurements-for-Call-of-Duty-Advanced-Warfare/ba-p/10730743
- [CIE 2006] CIE. 2006. Fundamental chromaticity diagram with physiological axes. Parts 1 and 2. Technical Report 170. Central Bureau of the Commission Internationale de l' clairage.
- [College 2013] OpenStax College. 2013. Rods and Cones.jpg. (2013). https://commons.wikimedia.org/wiki/File:1414 Rods and Cones.jpg
- [Debevec and Malik 1997] Paul E. Debevec and Jitendra Malik. 1997. Recovering High Dynamic Range Radiance Maps from Photographs. In Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'97). ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 369–378. DOI:http://dx.doi.org/10.1145/258734.258884

References

- [Fry 2017] Alex Fry. 2017. HDR COLOR GRADING AND DISPLAY IN FROSTBITE. (2017). https://www.ea.com/frostbite/news/high-dynamic-range-color-grading-and-display-in-frostbite
- [Grassman 1853] H. Grassman. 1853. Zur Theorie der Farbenmischung[DE]. Annalender Physik und Chemie. 165 (1853), 69–84.
- [HGIG 2018] HDR Gaming Interest Group. 2018. For Better HDR Gaming. (2018).http://hgig.org/
- [Hable 2010] John Hable. 2010. Uncharted 2: HDR Lighting. Game Developers Conference(2010).
- [Hallett 1969] P.E. Hallett. 1969. The variations in visual threshold measurement. J. Physiol. (Lond.) 202, 2 (Jun 1969), 403–419.
- [Hatada 1985] Toyohiko Hatada. 1985. Mechanism of Color Vision and Chromaticity Diagram[Translated from Japanese]. Bulletin of the Japanese Society of Printing Science and Technology 23, 2 (1985), 63–76. DOI:http://dx.doi.org/10.11413/nig1958.23.63
- [Inductiveload 2007] Inductiveload. 2007. EM Spectrum Properties edit.svg. (2007).https://commons.wikimedia.org/wiki/File:EM_Spectrum_Properties_edit.svg

References

- [Judd 1951] D.B. Judd. 1951. Report of U.S. Secretariat Committee on Colorimetry and Artificial Daylight. CIE Proceedings, (part 7) 1 (1951), 11. https://ci.nii.ac.jp/naid/10024222032/
- [Kawase 2016] Masaki Kawase. HDR Output Theory and Practice by SiliconStudio. (2016). https://www.siliconstudio.co.jp/rd/presentations/files/CEDEC2016/cedec2016_sskk_hdr_kawase.pptx
- [Kenneth Mees 1954] C.E. Kenneth Mees. 1954. L. A. Jones and his Work on Photographic Sensitometry. IMAGE Journal of Photography of the George Eastman House 3, 5 (1954), 34–36.
- [Lagarde 2016] Sebastien Lagarde. 2016. An Artist-Friendly Workflow for Panoramic HDRI. (2016). https://blogs.unity3d.com/jp/2016/08/28/59924/
- [Lottess 2016] Timothy Lottess. 2016. Advanced Techniques and Optimization of HDR Color Pipelines. (2016). https://gpuopen.com/gdc16-wrapup-presentations/
- [Malin 2018] Paul Malin. 2018. HDR in Call of Duty. Digital Dragons (2018).
- [Mitsunaga and Nayar 1999] T. Mitsunaga and S. K. Nayar. 1999. Radiometric self calibration. In Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149), Vol. 1. 374–380 Vol. 1. DOI:http://dx.doi.org/10.1109/CVPR.1999.786966

References

- [Ohta 1997] N. Ohta. 1997. The Basis of Color Reproduction Engineering. Corona-sha. [Translated from Japanese]
- [OpenCV 2018] OpenCV. 2018. Camera calibration With OpenCV. (2018). https://docs.opencv.org/2.4/doc/tutorials/calib3d/camera_calibration/camera_calibration.html
- [Patry 2017] Jasmin Patry. 2017. HDR Display Support in Infamous Second Son and Infamous First Light (Part 2). (2017). http://www.glowybits.com/blog/2017/01/04/ifl_iss_hdr_2/
- [GPU Gems2] Matt Pharr and Randima Fernando. 2005. GPU Gems 2: Programming Techniques for High-Performance Graphics and General-Purpose Computation (Gpu Gems). Addison-Wesley Professional, Chapter 24. https://developer.nvidia.com/gpugems/GPUGems2/gpugems2 chapter24.html.
- [Pirenne et al. 1957] M.H. Pirenne, F.H. Marriott, and E.F.O'Doherty. 1957. Individual differences in night-vision efficiency. Spec Rep Ser Med Res Counc (G B) 294 (1957), 1–69.
- [Dolby 2014] Dolby Vision Research. Dolby Vision white paper. https://www.dolby.com/us/en/technologies/dolby-vision/dolby-vision-white-paper.pdf
- [Dolby 2016] Dolby Vision Research. ICtCp white paper. (2016).https://www.dolby.com/us/en/technologies/dolby-vision/ICtCp-white-paper.pdf
- [Colour 2018] Colour science. 2018. (2018). https://github.com/colour-science/colour

References

- [Sharpe et al. 2005] L.T. Sharpe, Andrew Stockman, Wolfgang Jagla, and Herbert Jagle. 2005. A luminous efficiency function, V*(), for daylight adaptation. Journal of Vision 5, 11 (2005), 3. DOI:http://dx.doi.org/10.1167/5.11.3
- [Sharpe et al. 2011] L.T. Sharpe, A. Stockman, W. Jagla, and H. Jagle. 2011. A luminous efficiency function, VD65* (), for daylight adaptation: A correction.Color Research and Application 36 (2011), 42–46.
- [Smith and Guild 1931] Thomas. Smith and John. Guild. 1931. The C.I.E. colorimetric standards and their use. Transactions of the Optical Society 33, 3 (1931), 73. http://stacks.iop.org/1475-4878/33/i=3/a=301
- [Stiles and Burch 1959] W.S. Stiles and J.M. Burch. 1959. N.P.L. Colourmatching Investigation: Final Report (1958). Optica Acta: International Journal of Optics 6, 1 (1959), 1–26. DOI:http://dx.doi.org/10.1080/713826267
- [Stockman and Sharpe 2000] A Stockman and L.T. Sharpe. 2000. The spectral sensitivities of the middle- and long-wavelength-sensitive cones derived from measurements in observers of known genotype. Vision Research 40 (2000), 1711–1737.
- [Stockman et al. 1999] A. Stockman, L.T. Sharpe, and C. Fach. 1999. The spectral sensitivity of the human short-wavelength sensitive cones derived from thresholds and color matches. Vision Research 39 (1999), 2901–2927.

References

- [Uchimura 2016] Hajime Uchimura. 2016. https://www.slideshare.net/nikuque/color-science-for-gamesjp [Translated from Japanese]
- [Uchimura 2017] Hajime Uchimura. 2017. HDR Theory and Practice [Translated from Japanese]. CEDEC (2017).
- [Vos 1978] J.J. Vos. 1978. Colorimetric and photometric properties of a 2 fundamental observer. Color Research and Application 3, 3 (1978), 125–128.
- [Wyszecki and Stiles 1982] G. Wyszecki and W.S. Stiles. 1982. Color Science: concepts and methods, quantitative data and formulae. New York: Wiley.
- [XRite 2018] XRite. 2018. Color Checker Passport. (2018). http://xritephoto.com/colorchecker-passport-photo