



本日の内容

- HDR 対応テレビとは?
- HDR 出力デモ
- HDR 出力への対応
 - 色空間の対応
 - 出力信号規格への対応
 - 魅力的な映像のための対応
- その他
 - HDR対応における留意点
 - HDR出力の確認
 - HDR / SDR 比較スクリーンショットの撮り方?
 - 今後の課題
- まとめ





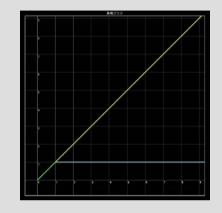
HDR 対応テレビ

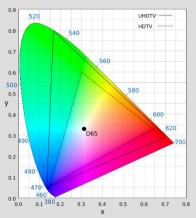
- ダイナミックレンジの広い映像を表示できる
 - HDR 規格に準拠した信号を入力して表示できる
 - 従来より暗い色と明るい色を表示できる



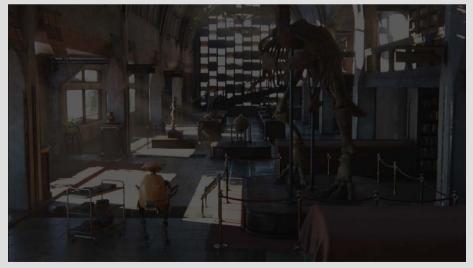
誤解を恐れず概念を簡単に言うと

- 1.0 を超える値を出力できる
 - 従来の最大輝度より明るい色
- 色域が広くなる
 - 従来よりも純色に近い色

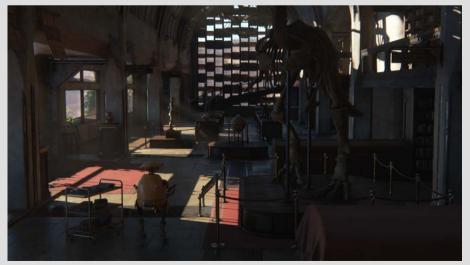




https://commons.wikimedia.org/wiki/File:CIExy1931 Rec 2020 and Rec 709.svg

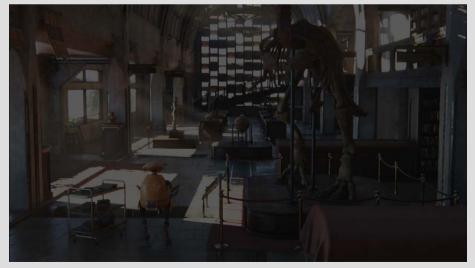




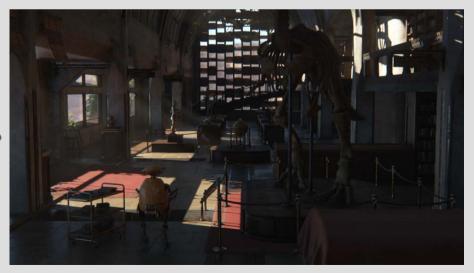


従来よりも高いダイナミックレンジ

- 暗い色と明るい色のコントラスト比が高い
- 特に最大輝度の高さが特徴
 - -×映像が明るくなる
 - ○明るい色も出せる





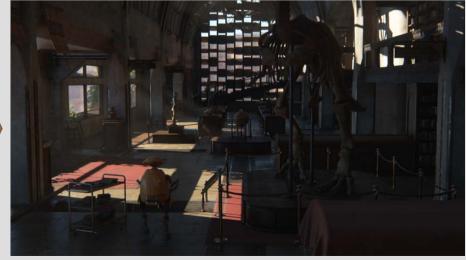


従来よりも高いダイナミックレンジ

- どのくらい明るいの?
 - SDR:数十~数百 nits (cd/m²)程度
 - HDR:数百~数千 nits (cd/m²) 程度以上







1.0を超える値の出力が可能

- 従来のモニタ(SDR)出力を 100nits であるとみなす
 - 従来は 1.0 (100nits)でクランプされていたと考える
- HDR の規格上は
 - 最大で 10 ~ 100 (1000~10000nits) までの輝度を出力できる
 - 実際の信号は正規化される























具体的に何を変える必要があるのか?

- 物理ベース等の正しい HDR レンダリング
 - 大前提
- 色空間の変更
 - 色域の変更
 - トーンマップの変更
 - ガンマカーブの変更
 - UI 対応
- 出力信号規格への対応
 - HDR メタデータの設定
- etc.
- ここでは特に重要な項目を抜粋して紹介
 - 基本的な内容から順に説明





EOTF (Electro-Optical Transfer Function)

- 従来とは異なる出力信号処理が行われる
 - SDR では主にガンマカーブ(モニタガンマ)が適用された
 - 2.2 / 2.4 / sRGB / BT.709 etc.
 - HDR はガンマカーブとは異なる EOTF を利用
 - EOTF (Electro-Optical Transfer Function)
 - 入力信号から出力輝度への変換関数を一般化した呼び方
 - 従来のいわゆるモニタガンマに相当



HDR 方式

- 大きく2種類の方式が標準化されている
 - ARIB STD-B67: HLG (Hybrid Log-Gamma)
 - SMPTE ST-2084: PQ (Perceptual Quantization)
- コンテンツ側は逆関数でエンコードした信号を出力
 - 従来のいわゆるガンマ補正に相当
 - OETF (Opto-Electronic Transfer Function) で非線形変換



どのような変換か?

- HLG (Hybrid Log-Gamma)
 - (日)NHK / (英)BBC が共同で開発
 - 従来のガンマカーブと対数カーブの合成関数
 - システムガンマにもよるが 20倍(2000nits 相当)程度まで出力可能(未確定)
 - 相対輝度
 - 従来のガンマカーブとの互換性が比較的高い
 - 従来と同様の圧縮方式でのクォリティが比較的高い
- PQ (Perceptual Quantization)
 - (米)Dolby が開発
 - 人間の視覚特性に最適化されたカーブ
 - ダイナミックレンジが非常に広い(最大 10000nits)
 - 絶対輝度
 - 従来のガンマカーブとの互換性が低い
 - SDR モニタに出力すると非常に不自然な色に!
 - 従来と同様の圧縮方式ではクォリティが非常に低くなる



HDR-TVにおける伝達関数

HLG (Hybrid Log-Gamma)方式 PQ (Perceptual Quantization)方式 $E' = r\sqrt{L}$ $(0 \le L \le 1)$ $E' = \left(\frac{c_1 + c_2 L^{m_1}}{1 + c_2 L^{m_1}}\right)^{m_2}$ $(0 \le L \le 1)$ $E' = a \cdot \ln(L - b) + c \qquad (1 < L)$ ただし、rは基準白レベルに対する映像信号レベ ただし、Lはカメラの入力光に比例した電圧とし、 ルでありr = 0.5とする。Lは基準白レベルで正規 L = 1が表示輝度10,000 cd/m²に対応するものと 化したカメラの入力光に比例した電圧とし、E'は する。E'は映像信号のカメラ出力に比例した電 映像信号のカメラ出力に比例した電圧とする。a, 圧とする。 m_1, m_2, c_1, c_2, c_3 は定数であり、以下 b, cは定数であり、以下のとおりとする。 のとおりとする。 $m_1 = 2610/4096 \times \frac{1}{4} = 0.1593017578125$ a = 0.17883277 $m_2 = 2523/4096 \times 128 = 78.84375$ b = 0.28466892 $c_1 = 3424/4096 = 0.8359375 = c_3 - c_2 + 1$ c = 0.55991073



 $c_2 = 2413/4096 \times 32 = 18.8515625$ $c_3 = 2392/4096 \times 32 = 18.6875$

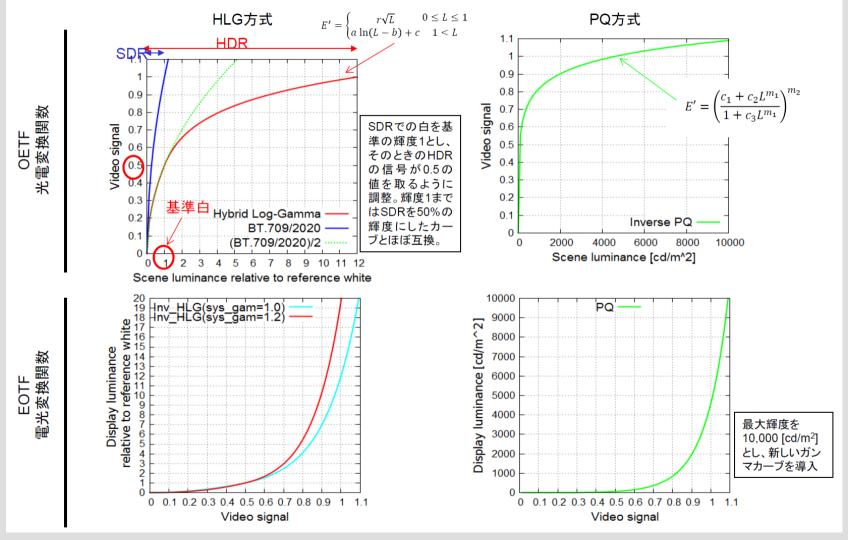
OETF の内容

- HLG (Hybrid Log-Gamma) の OETF
 - OETF 側を規定し、EOTF はその逆変換(EOTF-1)として定義
 - 0.0~12.0 を 0.0~1.0 に非線形変換
 - ただしシステムガンマ 1.2 のため(後述)
 - 0~20 程度までの輝度情報を 1/1.2 乗でガンマ補正
 - 0~12 の入力にしてから OETF を適用
- PQ (Perceptual Quantization) の OETF
 - EOTF 側を規定し、OETF はその逆変換(EOTF⁻¹)として定義
 - 0.0~100.0を0.0~1.0に非線形変換
 - 入力値の 1.0 を 100.0(10000nits) とみなしている
 - 最初に 1/100 で乗算してから OETF を適用



(参考)変換関数の比較(HLG方式とPQ方式)

8



情報通信審議会 情報通信技術分科会 資料より http://www.soumu.go.jp/main_content/000417993.pdf



システムガンマ(トータルガンマ)

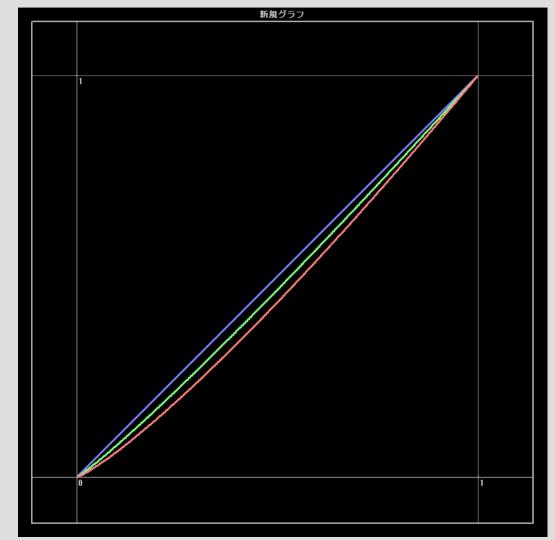
- 撮影時と鑑賞時とのトータルでのガンマ値
- 従来の SDTV(BT.601) / HDTV(BT.709) 規格の場合
 - カメラ側の OETF はガンマ 1.96~2.0 の補正
 - テレビモニタの EOTF はガンマ 2.2~2.4 程度
 - 実写映像のトータルガンマはおよそ 1.1~1.2 程度
 - 1.0 ではない(つまり線形ではない)
 - リアルタイムレンダリングの結果は sRGB 補正がメインなのでトータルほぼ 1.0
- 1.0 (線形)じゃないの?
 - 1.1~1.2 程度の値の方が自然(現実に近い)と感じる人が多い
 - 背景輝度やピーク輝度によって変えるべきという研究結果



ガンマの比較

1.0

1.2

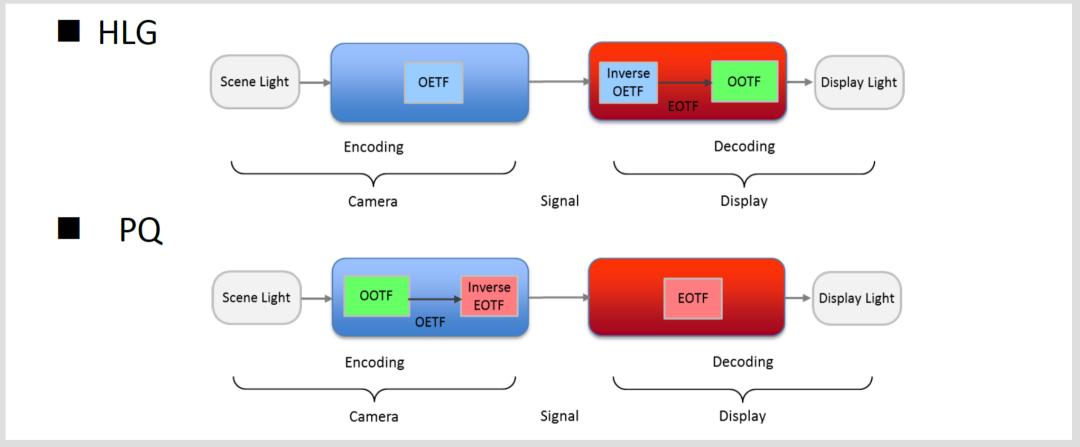


HDR におけるシステムガンマの扱い

- EOTF とは別にガンマカーブが適用される
 - OOTF (Opto-Optical Transfer Function) として一般化
- 適用タイミングは方式によって異なる
 - OETF を規定している方式(HLG)
 - OETF を規定し、EOTF はその逆関数(OETF-1)とする
 - EOTF を規定している方式(PQ)
 - EOTF を規定し、OETF はその逆関数(EOTF-1)とする
- 具体的にどう違うの?



OOTF (Opto-Optical Transfer Function)

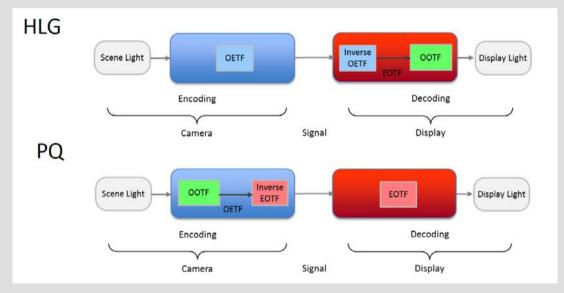


HDR-TVに関する ITU-R SG6会合(2016年1-2月)の結果より http://www.soumu.go.jp/main_content/000402269.pdf



OOTF (Opto-Optical Transfer Function)

- OOTF の適用タイミング
 - OETF を規定している方式(HLG)
 - ディスプレイ側で適用
 - EOTF (OETF-1) によるデコード後
 - EOTF を規定している方式(PQ)
 - カメラでの撮影時に適用
 - OETF (EOTF-1) によるエンコード前



HDR-TVに関する ITU-R SG6会合(2016年1-2月)の結果より http://www.soumu.go.jp/main_content/000417993.pdf



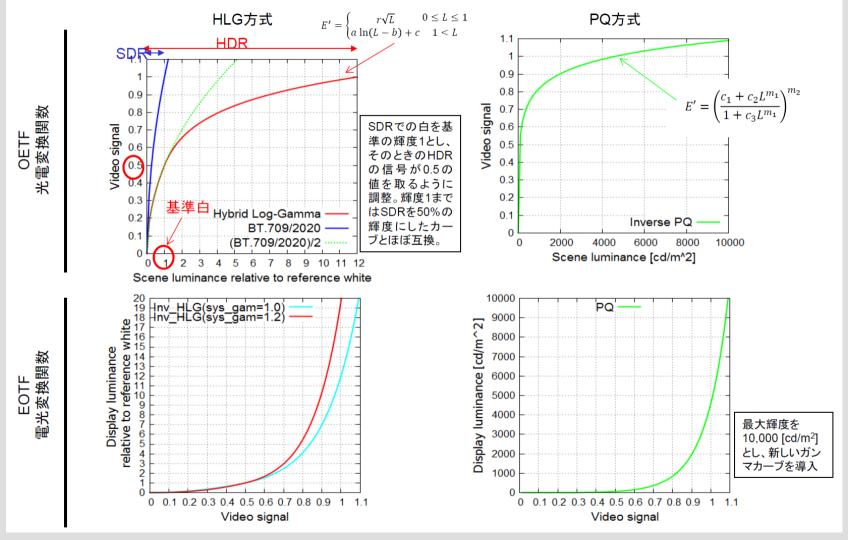
HLG ではシステムガンマの考慮が必要

- CGでのレンダリング結果の出力に影響がある
 - HLG
 - ディスプレイ側で EOTF の後に OOTF が適用される
 - レンダリング時と異なるガンマで表示される
 - PQ
 - カメラ側で OETF 前に OOTF が適用される
 - ディスプレイ側では適用されない
 - レンダリング時と同じガンマで表示される



(参考)変換関数の比較(HLG方式とPQ方式)

8



情報通信審議会 情報通信技術分科会 資料より http://www.soumu.go.jp/main_content/000417993.pdf

システムガンマの補正

- HLG では OETF 変換前にガンマ補正を行う
 - システムガンマ 1.2 の場合
 - 0~20 程度までの輝度を 1/1.2 乗でガンマ補正
 - 0~12.0 の入力にしてから OETF (EOTF-1) を適用
 - システムガンマは輝度信号に対して適用される
 - 厳密には YCbCr 空間で輝度信号を補正する必要がある
- 補正の結果トータルガンマは 1.0 となる
 - その上で必要に応じて任意のガンマを適用すると良い
 - 適用タイミングは OETF (EOTF-1) エンコードの前



仕様は完全には確定していない

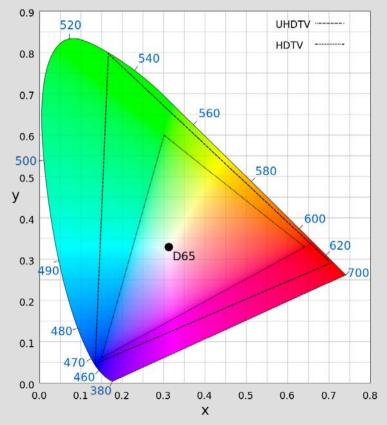
- とりあえず 1000nits 用に 1.2 と規定
 - 背景輝度やピーク輝度によって変えるべき
- ピーク輝度も最終決定されていない
 - 1000nits \sim 2000nits
 - システムガンマ適用後の 20 の明るさを何nitsとするか
 - 例えば 1000nits のディスプレイでは 20 は何nits?
 - 現在は 1000nits でクランプ
 - 複数の案によって最終的な表示輝度も異なる
- etc.
- 今後の動向によっては実装に変更が必要となる
 - ガンマ値をいくつにするか
 - どの明るさを基準にガンマを適用するか





色域の違いへの対応

- SDR では一般に BT.709
 - 色域は sRGB も同じ
- HDR では一般に BT.2020
 - BT.2020 が必須ではない

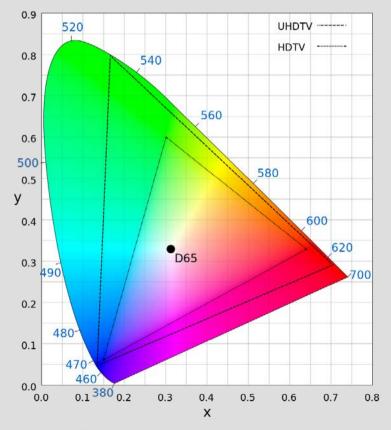


https://commons.wikimedia.org/wiki/File:CIExy1931_Rec_2020_and_Rec_709.svg



OETF 変換の前に色空間を変換

- レンダリングが sRGB 色域の場合
 - BT.709 (sRGB) から BT.2020 へ変換
 - トーンマップや OETF 適用前に
 - RGB 値の単純な一次変換
 - 以下の変換行列を作成して乗算
 - BT.709 を CIE XYZ 表色系に変換
 - CIE XYZ 表色系から BT.2020 に変換





https://commons.wikimedia.org/wiki/File:CIExy1931 Rec 2020 and Rec 709.svg

BT.709 から BT.2020 への変換行列

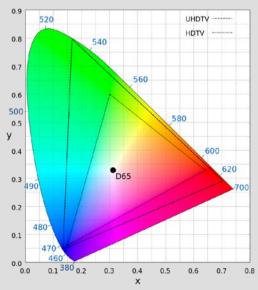
• 求め方によって係数も多少異なるため参考程度

$$\begin{pmatrix} R_{2020} \\ G_{2020} \\ B_{2020} \end{pmatrix} = \begin{pmatrix} 0.6275 & 0.3293 & 0.0433 \\ 0.0691 & 0.9195 & 0.0114 \\ 0.0164 & 0.0880 & 0.8954 \end{pmatrix} \begin{pmatrix} R_{709} \\ G_{709} \\ B_{709} \end{pmatrix}$$



最低限の広色域対応はこれで終了

- 既存の sRGB 色域をそのまま出力するだけ
 - 新しい規格の広色域を活かすことはできない
 - sRGB では表現できない自然界の色も多い
 - 広色域を活かすためにはより高度な対応が必要(後述)







HDR 規格に基づいた信号への対応

- 色空間の対応だけでは適切な HDR 出力はできない
 - ディスプレイ側が HDR 信号であることを認識できない
- 展示デモ等ではモニタ側を強制的に HDR モードに
 - HDR 用に EOTF / 色域等を設定できるモニタが必要
 - 正式な信号に対応していなくても表示可能
- HDR 対応コンテンツは自動的に HDR 表示されるべき
 - HDR 規格への対応が必要



HDR 規格に基づいた信号への対応

- HDR 規格への対応が必要
 - ハードウェア側
 - HDR 規格に対応したテレビ
 - ソフトウェア側
 - 規格に基づいた適切な HDR 信号を出力するための対応



HDR 規格(リアルタイム用途で利用可)

- HDMI 2.0a / DisplayPort 1.4
 - ST-2086 (Mastering Display Color Volume Metadata)
 - 色域 (原点および白色点座標)
 - 最大/最小輝度
 - 静的 HDR メタデータ
 - MaxCLL (Maximum Content Light Level: コンテンツ内の最大輝度)
 - MaxFALL (Maximum Frame Average Light Level: フレーム内平均輝度の最大値)
- Dolby Vision (HDMI 1.4 でも 4K HDR が可能)
 - 8-bit RGB を用いた 12-bit YCbCr 4:2:2 信号の挿入
 - 静的/動的 HDR メタデータ
 - 色域/シーン内の最大/最小輝度 etc.



HDR 規格信号の出力

- ディスプレイドライバの対応が必要
 - OS / GPU ベンダの対応
- HDR 対応のための API
 - GPU ベンダ独自の API が整いつつある
 - DirectX11/12
 - DXGI 1.5 で対応
 - その他のプラットフォーム
 - 専用 API 等が整いつつある



HDR 規格信号への対応(典型的なケース)

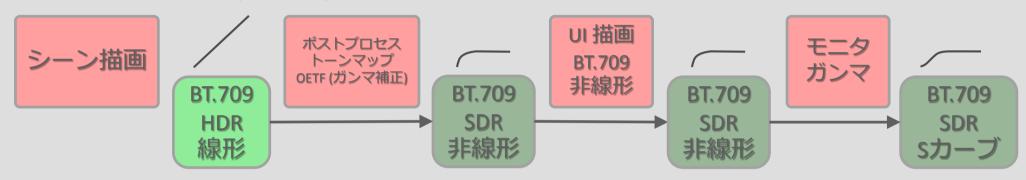
- おおまかな流れ
 - モニタの HDR 対応状況の取得
 - HDR 出力できるなら
 - HDR 出力設定のスワップチェーン作成
 - 可能なら固有 API で HDR メタデータを設定
 - HDR 出力用レンダリングフローに変更
- 詳細な仕様は各種 API ドキュメント等を参照



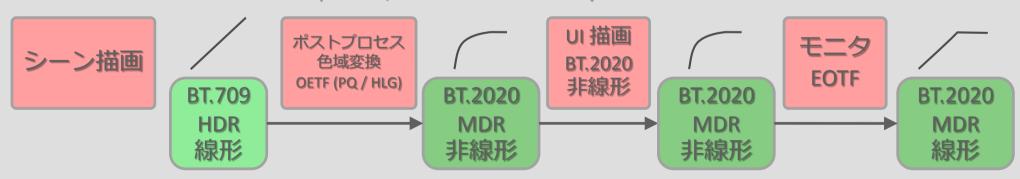


HDR レンダリングフロー(簡易な例)

SDR 出力のための(従来の)フロー



HDR 出力のためのフロー(OETF / 色域の変化のみ)





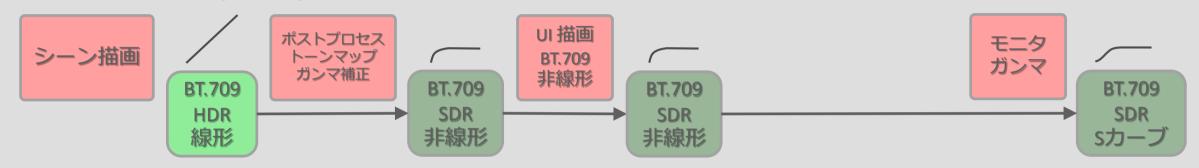
UIレンダリングの色空間

- 従来の SDR 出力
 - sRGB の非線形ガンマ空間に直接描画するケースが多かった
- HDR 出力でも従来どおり色空間/OETF 変換後に描画する場合
 - ターゲット色空間で描画しなくてはならない
 - HLG/PQカーブ
 - BT.2020
 - 色空間に起因する問題に注意
 - ブレンド/エッジアンチエイリアスが不適切
 - 非線形空間での 1.0 はモニタ上では最大輝度になってしまう
 - いずれの方式も 0.5 付近が従来の白(基準白レベル)となる
- 色空間/OETF 変換前に描画するように統一するのが無難
 - BT.709 線形空間で 1.0 を標準的な白の明るさとして描画

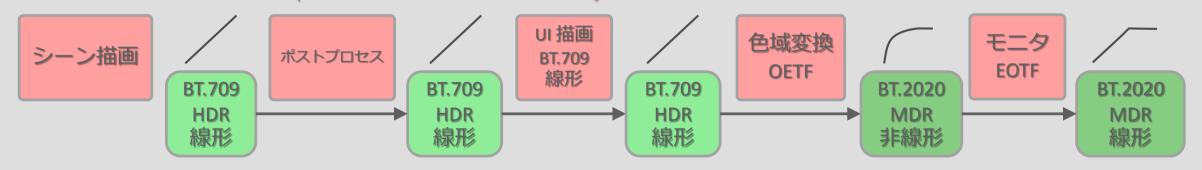


HDR レンダリングフロー(OETF 前に UI 描画)

SDR 出力のための(従来の)フロー



HDR 出力のためのフロー(UI を OETF 前に線形で描画)





Dolby Vision の場合

- HDR 用設定のスワップチェーンは不要
 - プラットフォーム固有の HDR 対応 API は必須ではない
 - レンダリングフローは他方式と同様
- Dolby Vision 規格信号をレンダリングフレームに挿入
 - シェーダでリアルタイムにエンコード
 - 全て動的に設定可能
 - 8-bit RGB に 12-bit YCbCr 4:2:2 信号をエンコード
 - 上部数ライン分のみメタデータを挿入



HDR 規格信号の出力対応メモ

- 色域は BT.2020 あるいは任意の RGB 原点と白色座標
- EOTF は現状すべて ST-2084 (PQ)
 - 今後は HLG 方式も利用できる可能性あり
- シーンレンダリングは出来れば R16G16B16A16_FLOAT 以上で
 - R11G11B10_FLOAT では一部 HDR の精度を活かし切れない
- バックバッファは R10G10B10A2_UNORM 以上で
 - Dolby vision 以外(必ず RGBA8_UNORM)
 - sRGB フォーマットは使わないこと
 - sRGB とガンマ 2.2 は少し異なるため 2.2 乗でガンマ補正しても一致しない
 - 特に HDR 出力時には差異が目立つ
- 原則としてフルスクリーンモード
 - バックバッファフォーマットと同じ精度に



HDR 規格信号の出力対応メモ

- Window モードでは適切に動作しない
 - OS あるいは ウィンドウシステムによるサポートが必要
 - HDR 対応ビューポート以外の色が不適切に表示される
 - sRGB 情報を HDR 表示するため非常に眩しく高彩度になってしまう
- Dolby Vision ではビデオ設定に注意
 - sRGB/ガンマ設定 etc.
 - Dolby vision 規格の信号が破壊されテレビ側で認識できない
 - バッファのスケーリング等にも注意





HDR 出力規格への対応だけで充分か?

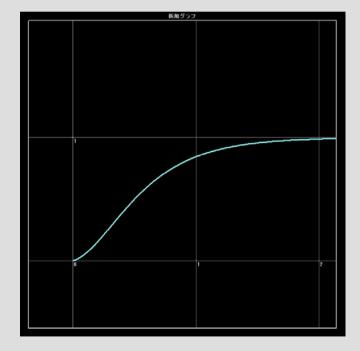
- ここまでの方法で HDR 対応テレビに出力は可能
 - 必ずしも魅力的な映像とは限らない
- 魅力的な映像のための対応
 - トーンマップとフィルミックカーブ
 - 基準白レベルの調整
 - 広色域レンダリングと色域マッピング
 - etc.





HDR でトーンマップは必要か?

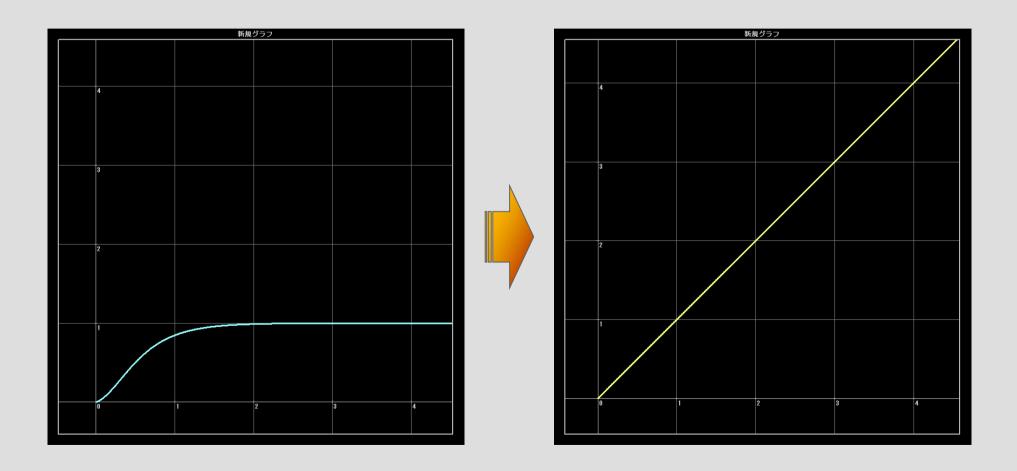
- 従来は HDR から SDR(LDR) への変換を行っていた
 - 0.0~1.0 への圧縮
- HDRではHDRからHDRへの変換?
 - 本来トーンマップは必要ない?



YEBIS デフォルト設定のフィルミックトーンカーブ



線形 HDR をそのまま表示できる?





「線形 HDR をそのまま表示」は理想論

- HDR モニタも当面は 1000~2000nits 程度まで
 - 10~20倍
- 10~20 程度でのクランプでは依然不自然な白飛び
 - 自然界をそのまま再現するためには全く足らない



メタデータを利用した ディスプレイ側の非線形飽和処理はどうか?

- ディスプレイの最大輝度によって入力値を飽和可能
 - MaxFALL / MaxCLL メタデータから
 - 民生用テレビでは独自に最適化される機種がある
 - Dolby Vision メタデータから
 - Dolby Vision 規格に基づいて最適に表示される
 - BT.2390 報告
 - ディスプレイ側の EETF (Electrical-Electrical Transfer Function) 変換
 - EOTF デコード前に PQ 空間等で非線形飽和
 - いずれは HDR テレビがサポートするかも知れない



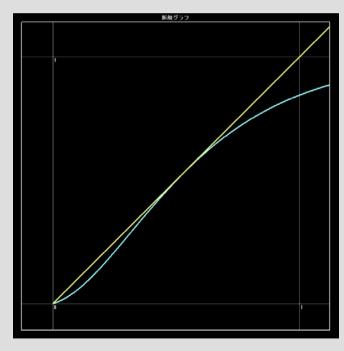
ディスプレイ側の非線形飽和処理は?

- 現状必ずしも適切に飽和されるとは限らない
 - コンテンツ側の意図通りに飽和されない
 - テレビ製品によって飽和のされ方が異なる



アーティスティックな効果

- フィルミックトーンカーブ(Sカーブ)など
 - 線形とは異なる絵作り
 - 高輝度出力問題とは別に必要な要素



線形とフィルミックカーブの違い



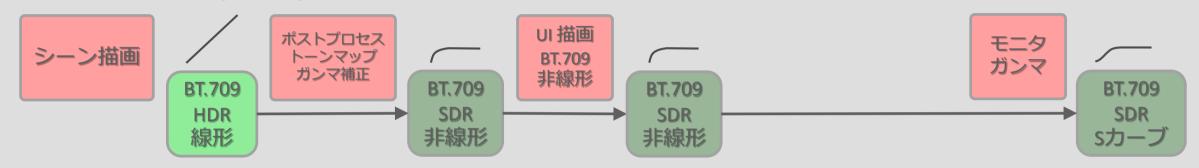
HDR でトーンマップは必要か?

- 非線形マッピングは必要
 - モニタ最大輝度への意図通りの飽和
 - フィルミックカーブ等のアーティスティックな調整
- HDR から MDR へのトーンマップと考えるとよい

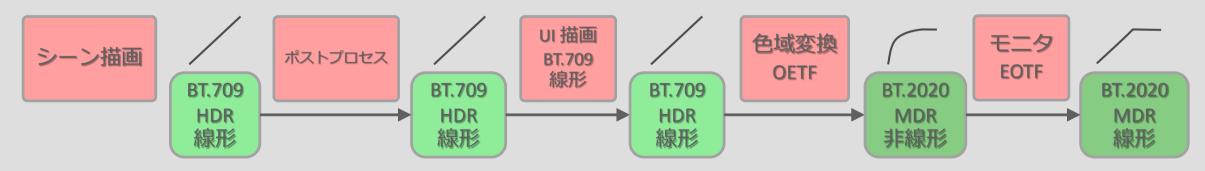


HDR レンダリングフロー(トーンマップ無し)

SDR 出力のための(従来の)フロー



HDR 出力のためのフロー

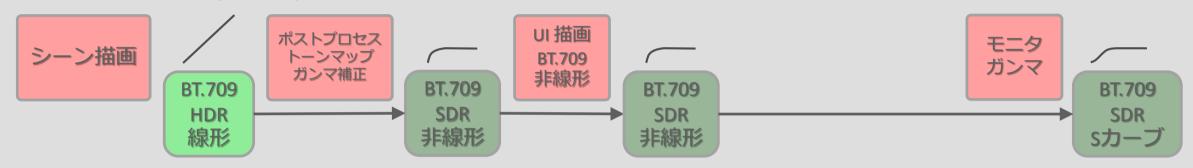




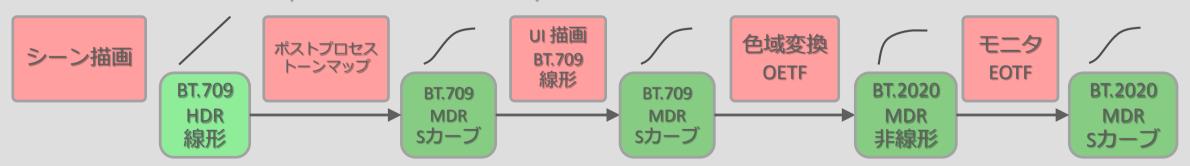


HDR レンダリングフロー(トーンマップ追加)

SDR 出力のための(従来の)フロー



HDR 出力のためのフロー(MDRトーンマップあり)





- HDR モニタの最大輝度は製品によって異なる
 - 従来の SDR では常に 0~1 への圧縮でよかった
 - HDR 対応では SDR を含めたあらゆるレンジへの圧縮が必要
 - 最大輝度ごとに最適化したトーンカーブが必要

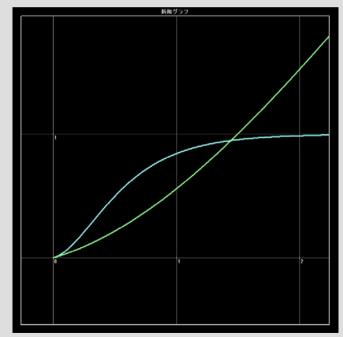


ディスプレイ最大輝度の取得

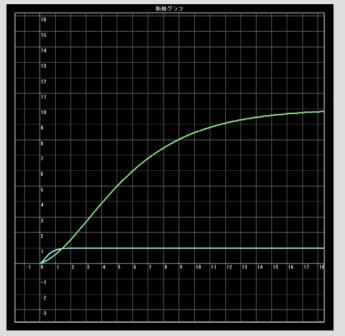
- ディスプレイの最大輝度が取得できるとは限らない
 - API が問い合わせをサポートしていない環境がある
 - API がサポートしていても数値が返るとは限らない
 - 現状では取得できないと考える方が無難?
- キャリブレーション
 - 実際に表示してユーザに判定させるしか無い?
 - 変化が見られない輝度を最大値として判断



- 試しにSカーブを最大輝度用に拡大してみる
 - 例えば最大輝度が 10 (1000nits) の場合単純に 10倍に拡大



映像の支配的な明るさ付近の拡大図

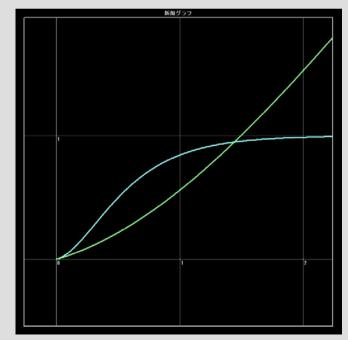


YEBIS のトーンカーブを 1000nits 用に拡大

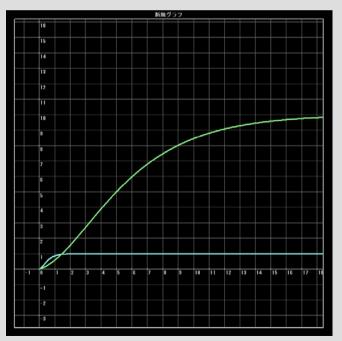
水色:SDR 用Sカーブ 緑色: HDR 用Sカーブ



- SDR 映像時の支配的な明るさ(18%グレイ付近)が大きく変化
 - 重要な映像部分のSカーブの特性が SDR 時と大きく異なる
 - SDR 時に意図した絵作りと一致しない
 - HDR 出力すると、重要な映像部分が SDR よりも不自然に暗くなる
 - 実質的にダイナミックレンジも狭くなってしまう



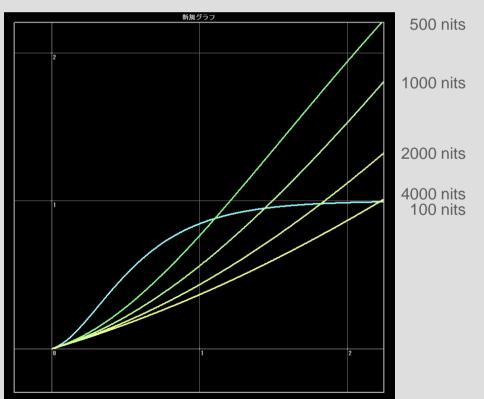
映像の支配的な明るさ付近の拡大図

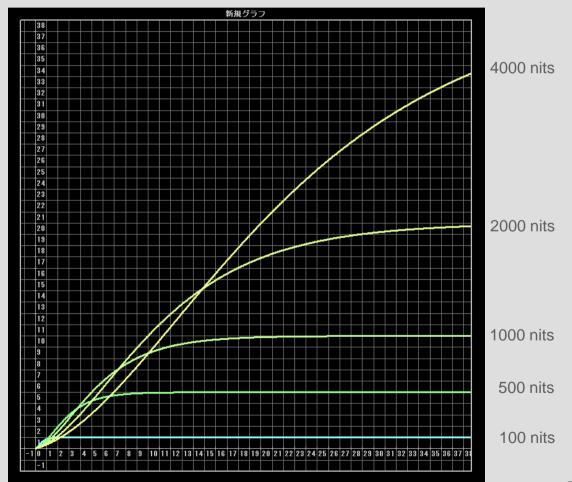


YEBIS のトーンカーブを 1000nits 用に拡大



- 最大輝度が高くなるとさらに悪化
 - 最大輝度毎に調整が必要になる?



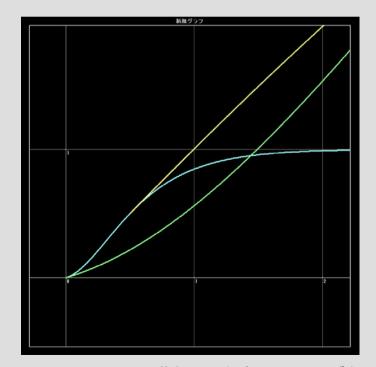


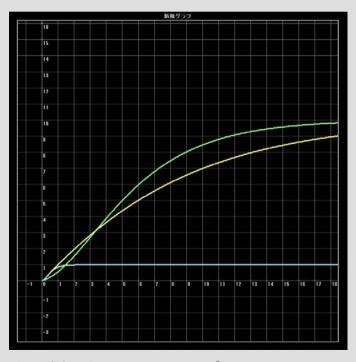
- 非常に好ましくない...
 - Sカーブの特性による一貫した絵作りをしたい
 - 最大輝度毎に絵作りが異なることは避けたい
 - 最大輝度毎にパラメタを調整することは避けたい
 - HDR 化に際し可能な限りパラメタを増やさない



HDR 出力のためのトーンマップ

- 0.5 程度以下の暗い部分を SDR と共通化
 - 0.5 程度以上の部分を最大輝度によって滑らかに飽和させる
 - 適正露出で出力した際の明るさが SDR とほぼ一致する



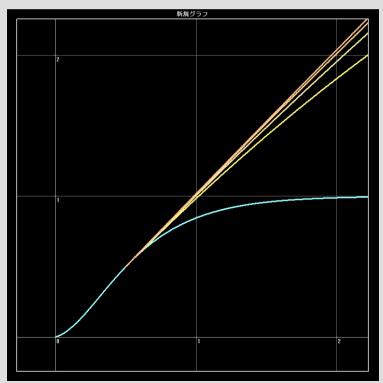


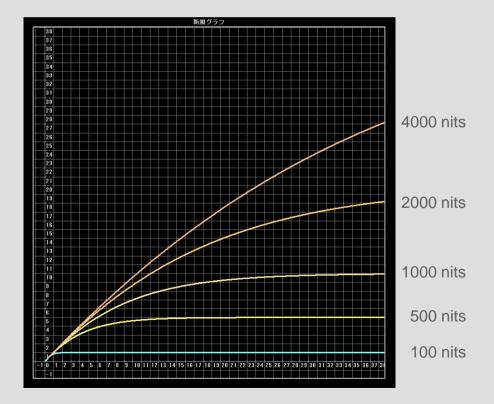


黄色:0.5 程度以下のカーブを SDR と一致させたHDR トーンマップ

HDR 出力のためのトーンマップ

- SDR を含めたあらゆる HDR 最大輝度に対応できる
 - 一貫したパラメタをすべての最大輝度に使用できる
 - SDR だけでパラメタを調整しても不自然な絵作りになりにくい

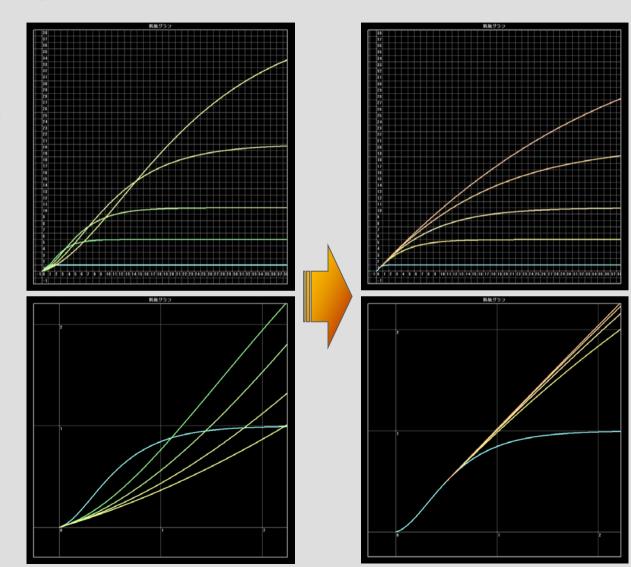






HDR 出力のためのトーンマップ

• SDR イメージを崩さずに より明るい部分を表現できる





シリコンスタジオデモの HDR 対応

- HDR 向けのパラメタの再調整は一切行っていない
 - 全てのパラメタは従来の SDR 用設定と共通
- 物理ベースレンダリングであることが強み
 - もともと不自然なパラメタが少ない
 - ダイナミックレンジが高くなっても破綻しない







露出調整における注意

- トーンマップは従来の数十倍の輝度まで出力できるが...
 - 映像の露出は変化させないことが基本
- 全体を明るく出力してしまっては HDR の効果が薄れる
 - 18%グレイ付近の映像は SDR 出力時と同じ明るさであること
- あくまで原則は...
 - では現実的には?





HDR 出力は規格通りに行うべきなのか?

- SDR テレビは本当に 100nits か?
 - CRT 時代の暗いディスプレイが 80~100nits 程度
 - 昼間は見難い程度に暗い
 - 近年の LCD は数百nits 程度
- 多くの民生用テレビは映像にかなり手を加えている
 - よりヴィヴィッドに
 - よりハイコントラストに
 - より高い色温度に
- sRGB 等の従来の SDR 規格にはあまり従っていない



HDR 出力は規格通りに行うべきなのか?

- 規格通りに HDR 出力すると?
 - SDR よりも暗くなってしまう
 - SDR よりも色が地味になってしまう
 - 広色域対応のはずなのに!
 - 民生用 HDR テレビの動向次第
 - HDR でもテレビ側で映像が大きく変更される可能性



SDR 出力に合わせた調整が必要?

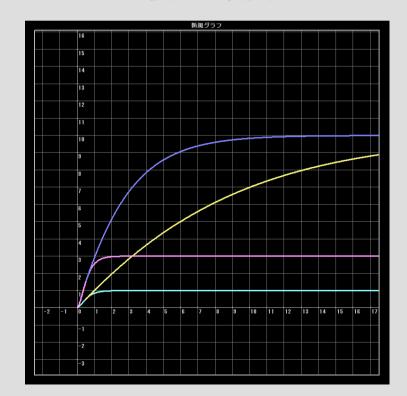
- SDR 時モニタ輝度を 100nits より明るいとみなす
 - 基準白レベル(Paper White Level)
 - SDR から HDR に変更した際に違和感を感じない程度に調整
 - SDR 表示時の白(最大輝度)の実際の明るさを基準白レベルに設定
- その他色域やコントラスト、色温度等
 - 従来から行われていた調整で可能
 - カラーグレーディングでの調整など

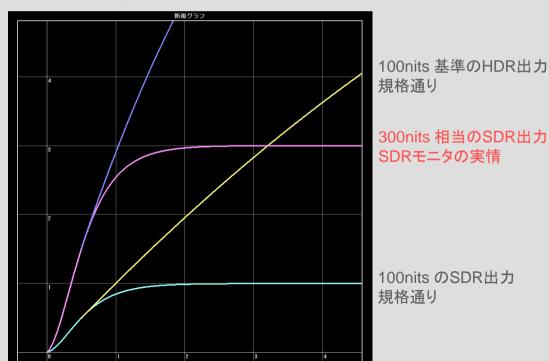


SDR 出力に合わせた白レベルの調整

- 例えば以下のようにみなした場合
 - SDR モニタ輝度(基準白レベル)が 300nits
 - HDR モニタの最大輝度が 1000nits

300nits 基準のHDR出力 実情に合わせた調整







SDR 出力に合わせた調整の注意

- 基本的には推奨しない
 - HDR 対応とは明るくすることではない
 - HDR 効果が低くなるので上げ過ぎないように
 - せいぜい 3倍(300nits)程度に
 - それでも HDR 効果は 1/3 になってしまう
- 露出を上げてしまわないように
 - トーンマップ前の線形空間でスケール(つまり露出変更)しない
 - 露出のスケールでは SDR 時とSカーブの互換性が失われる
 - トーンマップ後(OETF エンコード前)の非線形値をスケールする
 - トーンマップ時の飽和最大輝度を 1/3 にスケール
 - トーンマップ後のSカーブをそのまま 3倍にスケール

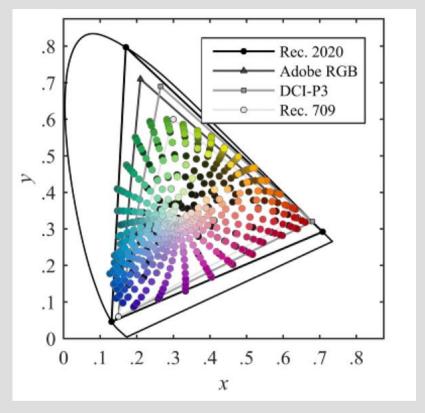






広色域の世界

- 自然界の色は BT.709 (sRGB) では表現しきれない
 - 鮮やかな鳥や花の色
 - 炎などの色温度
 - レーザー光
 - etc.
- BT.709 はポインターカラーの 74.4% をカバー
 - 実在する表面色のデータベース(右図は D65 光源下)
 - BT.2020 では 99.9% をカバー
 - ただし任意波長の単波長色は三原色では表現不可能



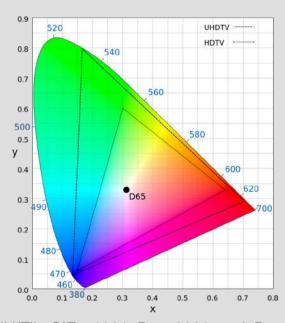
Pointer's colors under illuminant D65.

Kenichiro Masaoka, Yukihiro Nishida, "Metric of color-space coverage for wide-gamut displays," Opt. Express 23, 7802-7808 (2015); https://www.osapublishing.org/oe/abstract.cfm? URI=0e-23-6-7802



非均等色空間

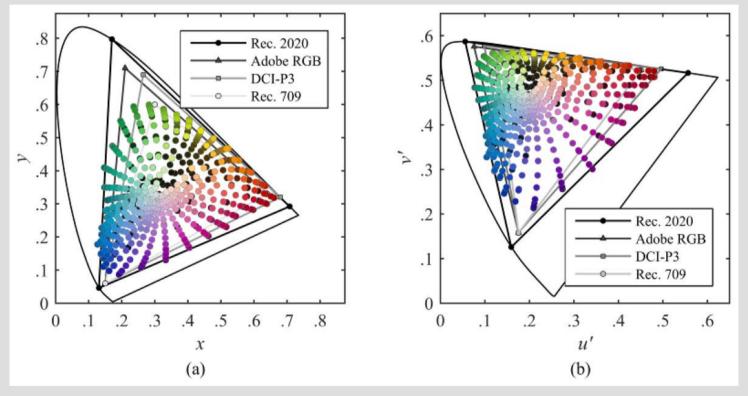
- CIE 1931 xy 色度図は均等色空間ではないので注意
 - 人間の知覚には非線形
 - 色度図上の面積や距離は知覚上の差異とは大きく異なる
 - CIE 1976 UCS (u'v') 色度図は比較的均等な色空間



0.7 0.6 0.3 0.1 0.3 0.5

https://commons.wikimedia.org/w/index.php?curid=173591SA 楕円の大きさが違いを知覚できる距離の10倍を表す

CIE 1931 xy と CIE 1976 u'v' 色度図上の D65 光源下ポインターカラー



Chromaticities for Rec. 709, Adobe RGB, DCI-P3, and Rec. 2020 RGB primary sets, and Pointer's colors under illuminant D65: (a) the CIE 1931 xy chromaticity diagram, and (b) the CIE 1976 u'v' chromaticity diagram.

Kenichiro Masaoka, Yukihiro Nishida, "Metric of color-space coverage for wide-gamut displays," Opt. Express 23, 7802-7808 (2015); https://www.osapublishing.org/oe/abstract.cfm?URI=oe-23-6-7802



広色域レンダリングへの対応段階

- シーンレンダリングは BT.709 色域で
 - レンダリングは BT.709 で最後に BT.2020 に変換
 - BT.709 外の色域はまったく使用しない
 - レンダリングは BT.709 で最後に色域マッピング
 - 擬似的な広色域化



広色域レンダリングへの対応段階

- シーンレンダリングから BT.2020 色域で
 - レンダリングの段階で BT.2020 化を適用
 - テクスチャは従来通り sRGB でロード時に BT.2020 に変換
 - あるいは任意のリソースのみ BT.2020 で製作
 - あるいはジオメトリバッファのアルベドを擬似的に広色域化
 - プロシージャルエフェクトは BT.2020 でシミュレーション
 - レンダリングの全てのパイプラインを BT.2020 対応
 - テクスチャ/ライティングも含め全てを BT.2020 で設定



広色域レンダリングへの対応

- 全てのパイプラインを BT.2020 対応できるのは当分先
 - 広色域モニタが潤沢に必要
 - アーティストに色域への正しい理解が必要
 - テクスチャ圧縮時の精度の問題
 - 広色域リソースは相対的に精度が低下する



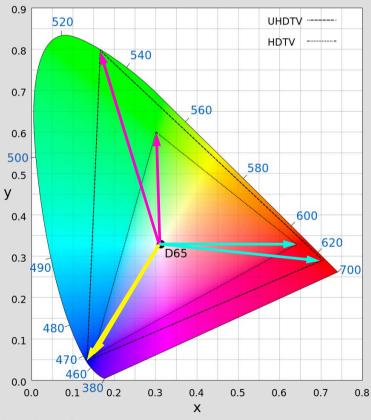
色域マッピング(Gamut Mapping)

- もっとも手軽な対応
 - パイプラインの変更が不要
 - あるいはジオメトリバッファのアルベドに適用
 - パイプラインの一部のパスの追加のみ
- 擬似的な色域拡張処理
 - 色域が制限されて BT.709 で飽和されてしまったと想定
 - 高彩度部分を非線形に拡張して元の色域を再現
 - あくまで擬似的な処理
 - 一部の広色域対応民生用テレビで行われているような処理



色域マッピングの難しさ

• 白色点から原色点への方向が異なる

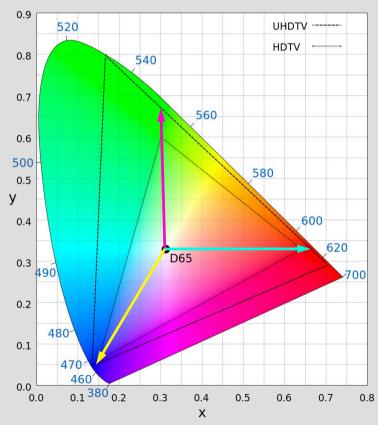


https://commons.wikimedia.org/wiki/File:CIExy1931_Rec_2020_and_Rec_709.svg



色域マッピングの難しさ

- 白色点から原色点への方向が異なる
 - 単純に彩度を上げる方向に拡張しても あまり理想的な広色域化ができない

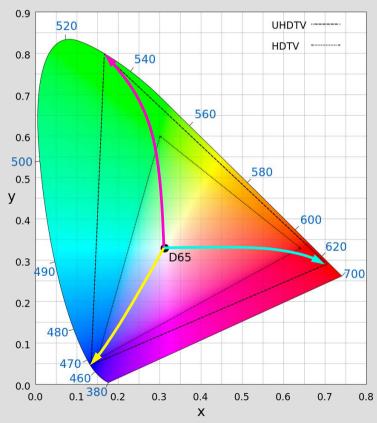


https://commons.wikimedia.org/wiki/File:CIExy1931 Rec 2020 and Rec 709.svg



色域マッピングの難しさ

- 白色点から原色点への方向が異なる
 - 単純に彩度を上げる方向に拡張しても あまり理想的な広色域化ができない
 - BT.2020 の原点方向に非線形に拡張
 - 色相の移動が発生してしまうので注意
 - 彩度の低い色には影響しないように注意
- しかしそもそも...

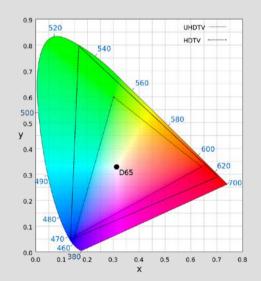


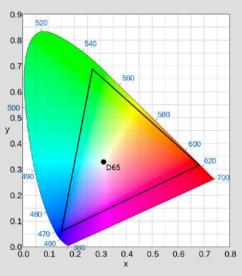




HDR テレビは BT.2020 を表示できるのか?

- BT.2020 の原色は純色(単波長色)
 - 630nm, 532nm, 467nm
 - つまりレーザーのような光源でしか表現できない色域
- 一般的な HDR テレビの色域は当面は DCI-P3 程度
 - デジタルシネマ規格の色域

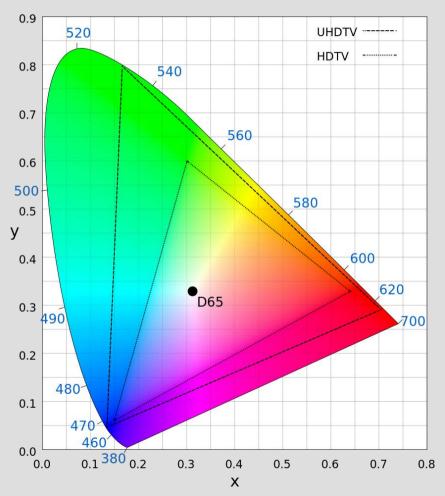


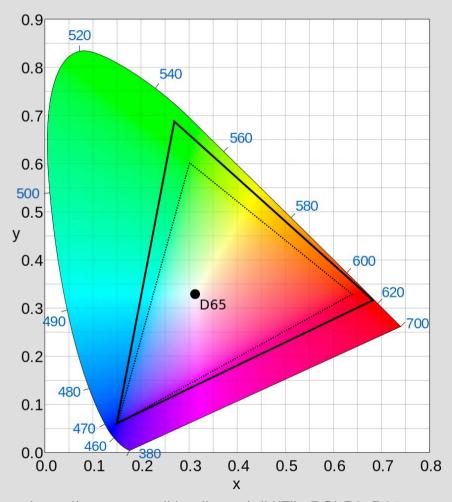


https://commons.wikimedia.org/wiki/File:CIExy1931_Rec_2020_and_Rec_709.svg https://commons.wikipedia.org/wiki/File:DCI-P3_D65.svg



BT.709 / BT.2020と DCI-P3 の色域





https://commons.wikimedia.org/wiki/File:CIExy1931_Rec_2020_and_Rec_709.svg

https://commons.wikipedia.org/wiki/File:DCI-P3 D65.svg



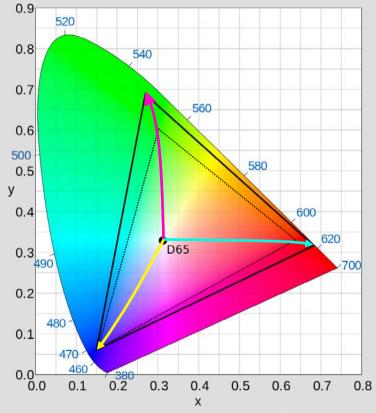
表示不可能な色域の飽和

- テレビで表示可能な色域を超える色は必ず飽和される
 - 方法は製品依存
 - 単純な彩度のクランプ
 - 非線形で滑らかな飽和
 - etc.
 - 彩度クランプされても輝度クランプ程粗は目立たない
 - しかしやはり高彩度部分の階調は失われるため避けたい



DCI-P3 程度に飽和

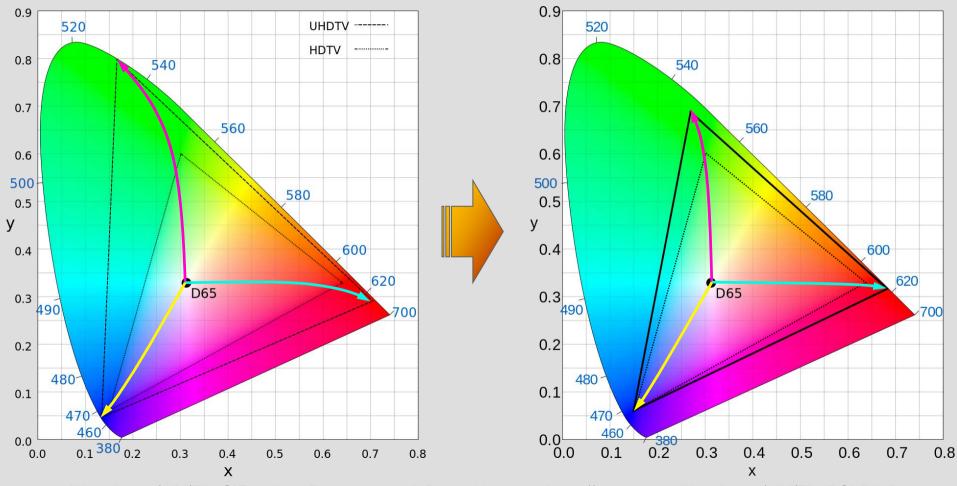
- DCI-P3 程度の色域で出力する方が自然で無難
 - 色域マッピングで DCI-P3 程度に飽和
 - BT.2020 ほど色相の移動が目立たない



https://commons.wikipedia.org/wiki/File:DCI-P3_D65.svg



BT.2020 と DCI-P3 の色域マッピングの比較





https://commons.wikipedia.org/wiki/File:DCI-P3_D65.svg



その他の小ネタ

- その他 HDR 対応における留意点
- HDR 出力の確認
- HDR / SDR 比較スクリーンショットの撮り方?
- 今後の課題





グレア/レンズフレア

- グレアやレンズフレアの調整は?
 - 物理的には設定を変更する必要はない
 - しきい値無しのシミュレーションに基づく場合
 - 多くのケースではアーティステックな調整が為されている
 - HDR ではやや弱めの設定に変更する方が無難



ポストプロセス系アンチエイリアス

- ポストプロセス系のアンチエイリアスの色空間
 - SDR (LDR) へのトーンマップを前提にした処理に注意
 - 1.0 を超える場所で適切な処理を行っていない可能性あり



ディザの有用性

- ディザリングを有効に
 - 10-bit PQ は若干精度が低い
 - 輝度によっては 8-bit sRGB と同レベル
 - Dolby Vision のように 12bits あれば充分
 - 量子化ビット数に最適化したディザリングが有効
 - OETF 適用後の非線形空間でオフセット
 - ディザを有効にすると 8-bit PQ でも何とかなる
 - トーンマップ後の最低/最大輝度にはオフセットしない
 - 特に明るい白にディザが適用されると目立ちやすい
 - SDR ではそれほど目立たなかったが HDR ではかなり目立つ



画面モードとバックバッファ

- 必ず同じ精度のフォーマットに
 - 10bits に量子化しても画面モードが 8bits では無意味
 - 精度が強制的に低下されてしまう
 - ディザも不適切なスケールとなる





シミュレーションモードの必要性

- HDR モニタはまだまだ普及には遠い
 - しかしアーティストは HDR 出力を確認する必要がある
- 当面は既存の SDR モニタで HDR を確認するしかない



シミュレーションモードの必要性

- SDR モニタで HDR 出力をシミュレーション
 - 例えばモニタ出力を 1000nits とみなして 10.0 で正規化
 - 本来の 1/10 の明るさで 10 倍までの輝度を sRGB 表示
 - SDR 出力とも比較可能に
 - SDR も HDR に白レベルを合わせて 1/10 輝度の表示モードを用意
 - 出来るだけ暗い部屋でモニタを明るくして確認
- 今回のスライドの比較画像はこの方法でキャプチャ
 - 本来 10 倍明るく表示されるべき画像







明るい SDR モニタでの HDR 表現

- 近年の明るいモニタへの SDR 出力はもったいない?
 - 明る過ぎるためにわざわざモニタを暗くする現状
- 画面を明るくして HDR シミュレーション
 - 500nits 程度以上の明るさがあれば実用可能
 - 出力スケールを任意に変更できると良い
 - 1/3~1/10 程度のスケールが実用的
 - 8-bit 出力では暗い領域の精度が不足
 - 酷いバンディングが発生する
 - ディザリングは必須





HDR / SDR 比較スクリーンショット

- 非常に難しい
 - 比較画像は SDR 環境で表示することが前提
 - そもそも不可能
 - SDR 表示で HDR の表現力と SDR との違いを説明
 - SDR 画像を通常よりも劣化させざるを得ない



比較スクリーンショットの作り方

- 写真で HDR 対応モニタを撮影
 - 普通に撮影して比較すると HDR が必ず白飛びしてしまう
 - 露出を抑えざるを得ないが SDR が暗くなり過ぎる
- この資料のような HDR シミュレーションで比較
 - やはり SDR が暗くなり過ぎる
 - 1000nits 想定なら本来の 1/10 の輝度
- 独自のレタッチを行う
 - ノウハウが必要
 - SDR / HDR の表現力の違いを正しく理解している必要あり



メーカによって比較方法もバラバラ

- 各社独自の方法で比較画像を作っている模様
- SDR / HDR の特徴が反対に見えるものも
 - コントラストを高くしたために HDR だけが白飛びしたり
 - 本末転倒



比較スクリーンショットの理想

- 出来るだけ SDR を劣化させない
 - 突っ込みどころとなる...
- HDR の強みを理解できるようにする
 - SDR では白飛びする明部の色の再現
 - ただ明るくなる訳ではない



輝度を 1/3 にしたシミュレーション

- SDR の明るさは 1/3 にして sRGB 出力
- HDR は 300nits 用トーンマップを 1/3 輝度で sRGB 出力
 - 基準白レベルを 33nits とした 100-nit HDR とも言える



輝度を 1/3 にしたシミュレーション

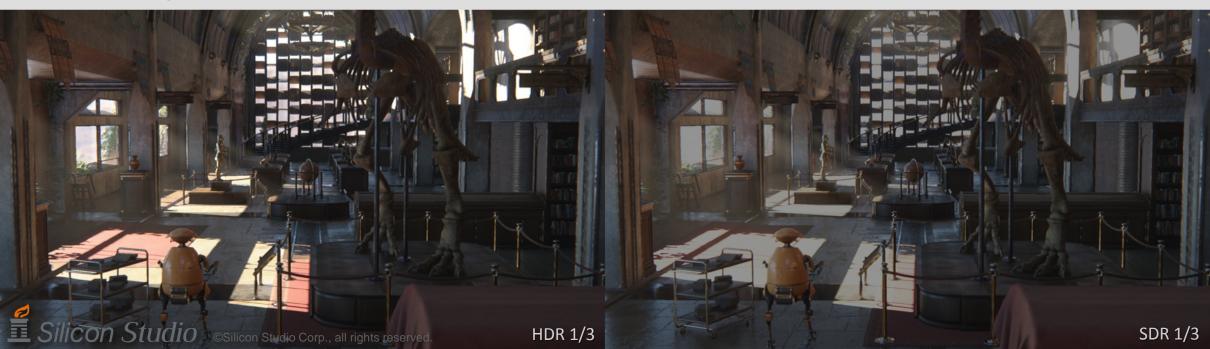
- HDR 画像
 - 明部の色味は 300-nit HDR を再現
- SDR 画像
 - 本来の 1/3 輝度となる
 - HDR シミュレーションの 1/10 よりも明るく劣化が少ない





輝度を 1/3 にしたシミュレーション

- SDR 画像はまずまず
 - 劣化させた感は少ない
- HDR にやや白飛びの傾向
 - 1/4 程度でも良いかも



HLG でエンコードしてみる

- HLG はもともと互換性の高さが特徴
 - 逆に擬似的な HDR 表示にも使えるはず
- 色域は BT.709 のまま HLG OETF を適用
 - これを SDR モニタで表示
 - SDR 画像は規定どおり 1.0 にトーンマップして HLG OETF
 - HLG の SDR 領域は BT.709 とほぼ互換



色域は BT.709 のまま HLG OETF を適用

- HDR 画像
 - 高輝度領域が対数圧縮された画像となる
 - 明部の色味は本当の HDR 程ではないが再現できる
- SDR 画像
 - 本来の 1/4~1/5 程度となる
 - 8-bit RGB 値で 127
 - HDR シミュレーションの 1/10 よりは明るく劣化が少ない





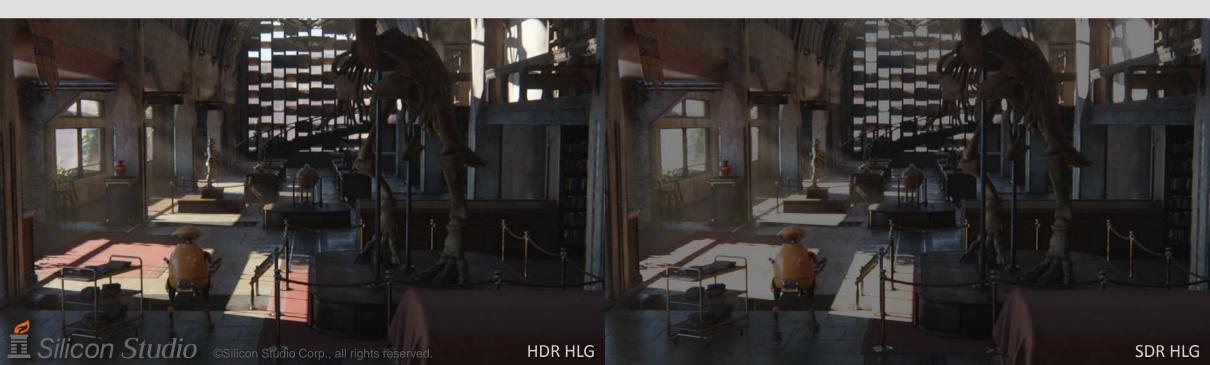






HLG でエンコード

- SDR 画像はやや暗め
- HDR 画像は飽和により高輝度の彩度がやや低め
 - 1/3 輝度のシミュレーションよりも白飛びは少ない





結果の評価

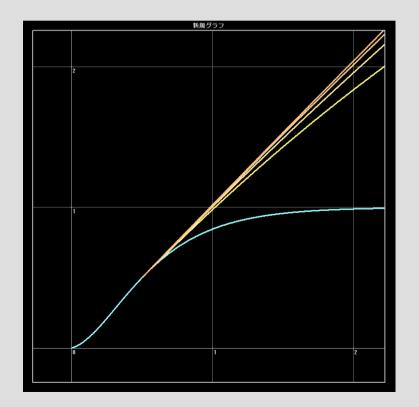
- いずれもまずます実用的
 - SDR 画像はまずまず良い感じ
 - 劣化させた感は少ない
 - HDR 画像はやや HDR 感が低い
 - ややSDRとの違いが少ないか?
 - 高輝度部分の圧縮や白飛びにより彩度が下がっている
 - OETF 前に彩度を保持するトーンカーブを適用すべき?
- トーンカーブや輝度スケールでいろいろと調整可能
 - お好みに合わせて調整してください

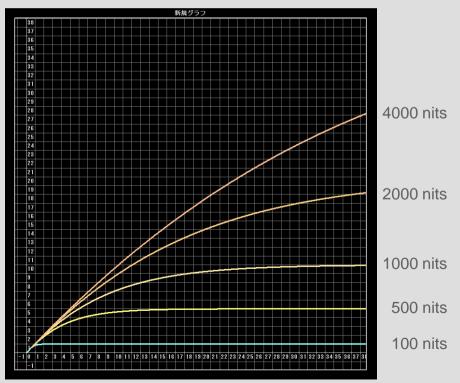




トーンマップの改善の余地

- 現状のシリコンスタジオのアプローチ
 - 互換性最優先







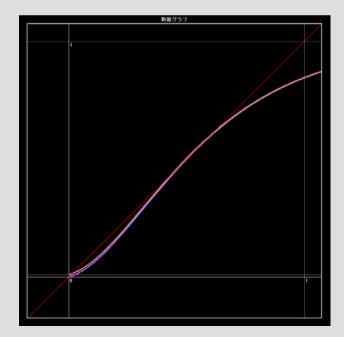
トーンマップの改善の余地

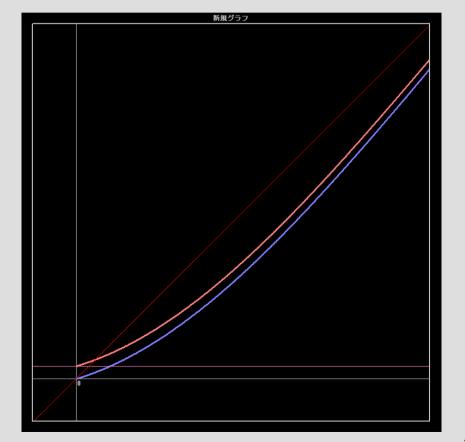
- HDR 用に最適化すべきか?
 - 各所からトーンカーブが提案されている
 - BT.2390, ACES, NVIDIA, AMD, etc.
 - 高輝度部分の飽和曲線
 - 彩度を維持して輝度のみカーブする方が良い?
 - 特に最大輝度が 1000nits 程度以下の場合
 - 高輝度部分の飽和をもっと弱くすべき?
 - 特に 5000nits 程度以上の領域
 - 暗部のSカーブ特性
 - HDR では黒もより締まる
 - SDR 時の Sカーブと同じでは黒が潰れ過ぎるかも?



暗部のSカーブ特性

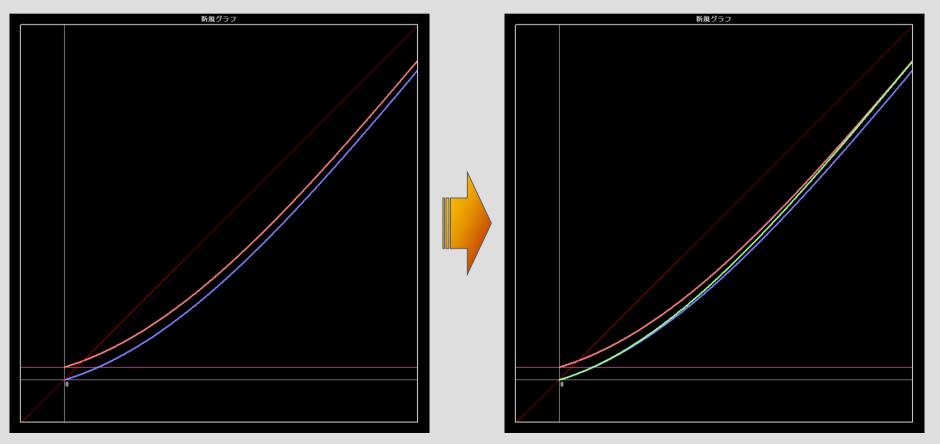
- HDR になると暗くなる?
 - 従来の SDR は黒浮きが大きい
 - 暗部が底上げされていたため





暗部のSカーブ特性

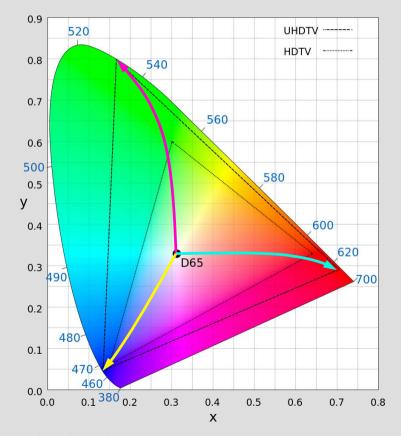
• 暗部のカーブを変更すべきか?





色域マッピングの改善の余地

- ICpCt 色空間での処理も提案されている
 - 色相の線形性が YCbCr 色空間よりも高い
 - RGB に戻した際にクランプされないか?
- 民生用テレビの方が高画質?
 - 大量のデータベースを元にしている
 - BT.709 のまま出力してテレビに任せる?
 - テレビ依存なので難しい
 - 一般視聴向けの設定ではヴィヴィッド過ぎる?





https://commons.wikimedia.org/wiki/File:CIExy1931 Rec 2020 and Rec 709.svg

- 本当に SDR と共通のパラメタで済ませられるのか?
 - HDR では明るい色が表示される
 - SDR では白飛びすることを前提にしていた部分も見えてしまう
 - HDR ではグレア/レンズフレアが強すぎる
 - SDR でアーティスティックな設定をしていた場合
- 基本は共通のパラメタとする
 - 必要に応じて一部のパラメタのみ HDR 用に別設定
 - 多くとも SDR / HDR の 2 種類程度にすること



- 明るく調整し過ぎに注意
 - HDR モニタが普及した際に問題になり得る
 - HDR モニタでパラメタ調整すると明るくしてしまいがち
 - HDR 対応とは明るくすることではない
 - 常に SDR 出力も確認しながら調整すること
 - SDR 時の白飛びを HDR で抑えられているか?
 - 比較モードを用意すると便利



- 適切な HDR(シーンリニア)情報の重要性
 - 空や太陽の輝度
 - 加算エフェクトの修正
 - 白飛びを利用したエフェクトは HDR 出力では破綻する
 - 物理的には高輝度のブレンドとすべき
 - トゥーンやノンフォト系などの物理に基づかない表現
 - 従来のものは基本的には互換性がない
 - 今後の開発は HDR 出力を視野に入れることが重要
 - 単純に明るくしてしまわないように注意



- ガンマ補正
 - 従来のガンマ補正をグレーディング用途に使っていた場合
 - HDR ではガンマ特性の大きな変更は実用的ではない
 - 明るい色が不適切な輝度になる可能性あり
 - トータルガンマで 0.9~1.2 程度までなら何とか



- LUT グレーディングの互換性
 - 従来の LUT はそのままでは使えない
 - HDR では 1.0 よりも高い値に拡張した LUT が必要
 - 自動拡張可能?
 - おそらく難しい
 - 初めから HDR を想定して作成する必要あり
 - HDR を想定した LUT は SDR にも適用可能
 - 格納方法は PQ や類似の非線形空間が提案されている





まとめ

- HDR 出力対応
 - EOTF 変換
 - 色域変換
 - HDR 規格信号の出力
 - HDR レンダリングフローの変更
- 魅力ある映像のために
 - トーンマップの変更
 - 色域マッピング
- HDR 画像比較
 - 輝度を下げたシミュレーション
 - 比較画像の作成
- 互換性の課題も...
 - パラメタは出来るだけ SDR と共通に
 - LUTのHDR対応



まとめ

- HDR の展望
 - SDR の 20倍以上の相対輝度が理想
 - 一般的なシーンなら非線形にトーンマップすれば充分表現可能
 - SDR でも 200~300nits 以上出る実情では 5000nits 以上が理想
 - 現状の 1000nits 程度ではまだまだ理想には遠い
 - たとえ 1000nits でも従来より表現の幅は広くなった
 - 映像を明るくするだけでなく表現の幅を広げる進化が可能
 - この認識を示せるようになってきたことは大きい





YEBIS、Mizuchi をはじめ 弊社ミドルウェアは 常に進化を続けています!







的影





References

- Recommendation ITU-R BT.601 "Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios." http://www.itu.int/rec/R-REC-BT.601/, 2011.
- Recommendation ITU-R BT.709 "Parameter values for the HDTV standards for production and international programme exchange." http://www.itu.int/rec/R-REC-BT.709/, 2015.
- Recommendation ITU-R BT.1886 "Reference electro-optical transfer function for flat panel displays used in HDTV studio production." http://www.itu.int/rec/R-REC-BT.1886/, 2011.
- Recommendation ITU-R BT.2020 "Parameter values for ultra-high definition television systems for production and international programme exchange." http://www.itu.int/rec/R-REC-BT.2020/, 2015.
- Recommendation ITU-R BT.2100 "Image parameter values for high dynamic range television for use in production and international programme exchange." http://www.itu.int/rec/R-REC-BT.2100/, 2016.
- Report ITU-R BT.2390 "High dynamic range television for production and international programme exchange." https://www.itu.int/pub/R-REP-BT.2390/, 2016.
- ARIB STD-B67 "ESSENTIAL PARAMETER VALUES FOR THE EXTENDED IMAGE DYNAMIC RANGE TELEVISION (EIDRTV) SYSTEM FOR PROGRAMME PRODUCTION ARIB STANDARD." http://www.arib.or.jp/tyosakenkyu/kikaku hoso/hoso std-b067.html, 2015.
- MSDN, "High Dynamic Range and Wide Color Gamut" https://msdn.microsoft.com/enus/library/windows/desktop/mt742103(v=vs.85).aspx
- NVIDIA GameWorks, "High Dynamic Range Display Development." https://developer.nvidia.com/high-dynamic-range-displaydevelopment

