

A 3D rendered character with short, dark hair, looking slightly to the right. The character is wearing a dark jacket. The background is a blurred, outdoor scene with some structures.

レンダリストのための物理ベースライティング

Physically Based Lighting for Rendering

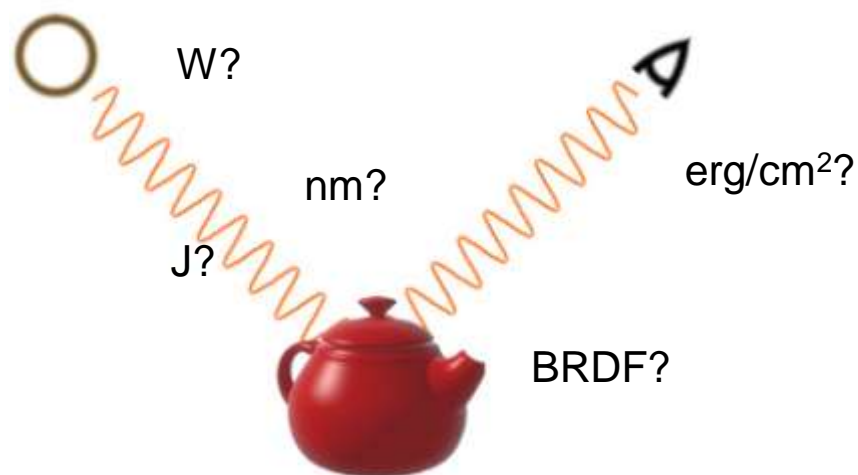
(株)トライエース 研究開発部
五反田義治

なぜ物理ベースライティング?

- BRDFのみの物理的正確性だけでは物理的に正しいレンダリングとは言えない...かも
 - Intensity 1.0ってどのくらいの光の強さ?
 - (1.0, 0.5, 0.1)って物理的にどんな色?
 - 100Wの白熱球はIntensityで言うと?
 - このシーンの露出はISO100でFいくつで何秒?
 - などなど...



- 物理ベースライティングと物理ベースでないライティングの違いとは
 - 正しい面積の考慮
 - 正しいエネルギー量の考慮
 - 光の波長(スペクトル)における取り扱い
 - 物理的なエネルギー量の取り扱い



- 面積については取り扱いません
 - 面積による見た目の効果は大きい
 - やわらかいライティング
 - 正確な影
 - 物理ライティングというよりは正しい(物理)レンダリングを行うということ



このセッションでは正しい光のエネルギー量について考察します

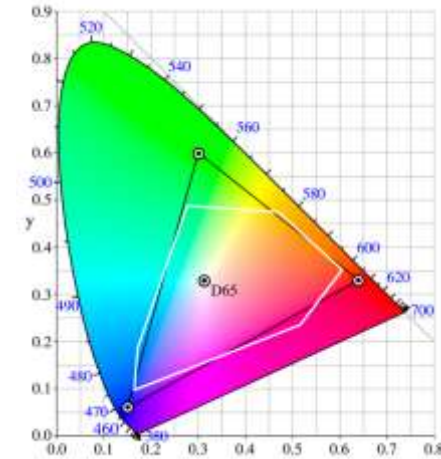
- レンダリング結果になにが影響するのか？

なにも変わりませんか？

実際には波長を考慮したレンダリングでは変化します

物理的特性および制限を考慮すれば変化します

- 物理ベースな設計自由度の高いライト
 - 幅広い色空間に対応が可能
 - 将来的な高色域レンダリングに対応
 - Adobe RGB
 - XYZ
 - xvYCC
 - スペクトルレンダリング
 - レンダリングと出力(sRGB)で異なる色空間
 - (3D LUT)トーンマップを考慮すればより自然に近い色表現
 - すでにsRGBより広い色域に対応したディスプレイがある
 - 将来は標準になるはず



Wikipediaより

- エネルギーベースのライト

- 正しい露出表現

- リアリティのあるカメラ、レンズ表現
 - デザイナーの設定が楽になる
 - いろいろなシチュエーションでも手動調整を最小限にする一貫したライティング

- 現実の光源を参考に行うことができる

- 太陽、空シミュレーション
 - 現実のライトなど各種資料を参考にする
 - 実際に測定
 - (HDR)イメージベースライティング



- 現実ではイメージセンサーの物理的な感度に限界がある
 - 100Wの光と1mWの光で照らされた物体は純粹に明るさが100,000倍違うだけだが...
 - デジタルセンサーであれば感度をあげればノイズが増える
 - 人間の眼では暗いところで彩度が落ちる
 - 錐体細胞
 - 桿体細胞

- 実際のカメラベースのシミュレーションを搭載
 - シャッタースピード、F値、ISO感度...
 - しかし正しい露出値はどのように決定するのか?
 - Sunny F16 Rule(快晴時F16の法則)
 - 快晴時の適正露出はシャッタースピード = $1 / \text{ISO感度}$ の時にF16になるというカメラの撮影テクニックの経験則
 - しかし物理的な値を持たないシーンやライティングでどのようにこの値を導き出すのか?



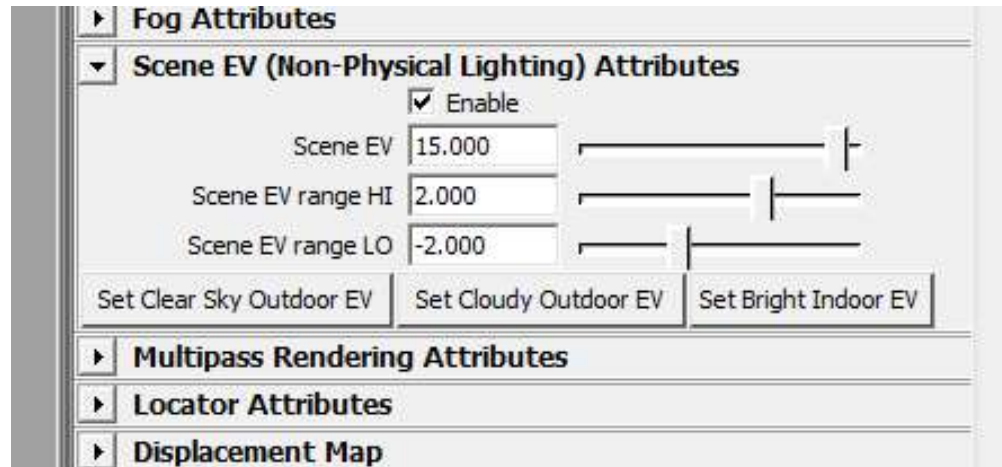
F2.8 1/8000 ISO200

F16換算で考えるとシャッタースピードは1/250



- カメラが判断する適正な露出とは？
 - 物質の反射率の平均を18%(グレー)と仮定
 - 被写体の明るさの(重み付け)平均を18%グレーになるように露出を調整
 - sRGBでは $0.18^{0.45} = 0.46$
 - 実際に平均は16%と言われている
 - ディフューズだけで考慮するとAlbedoが0.5
 - » $0.5/\pi = 0.16$

- 仮想露出パラメータ "Scene EV"
 - 非物理的なレンダリング結果を物理的な露出値と結びつけるための仮想パラメータ
 - デザイナーがシーンに手動で設定する
 - たとえば晴天時の屋外には"Scene EV = 15"など

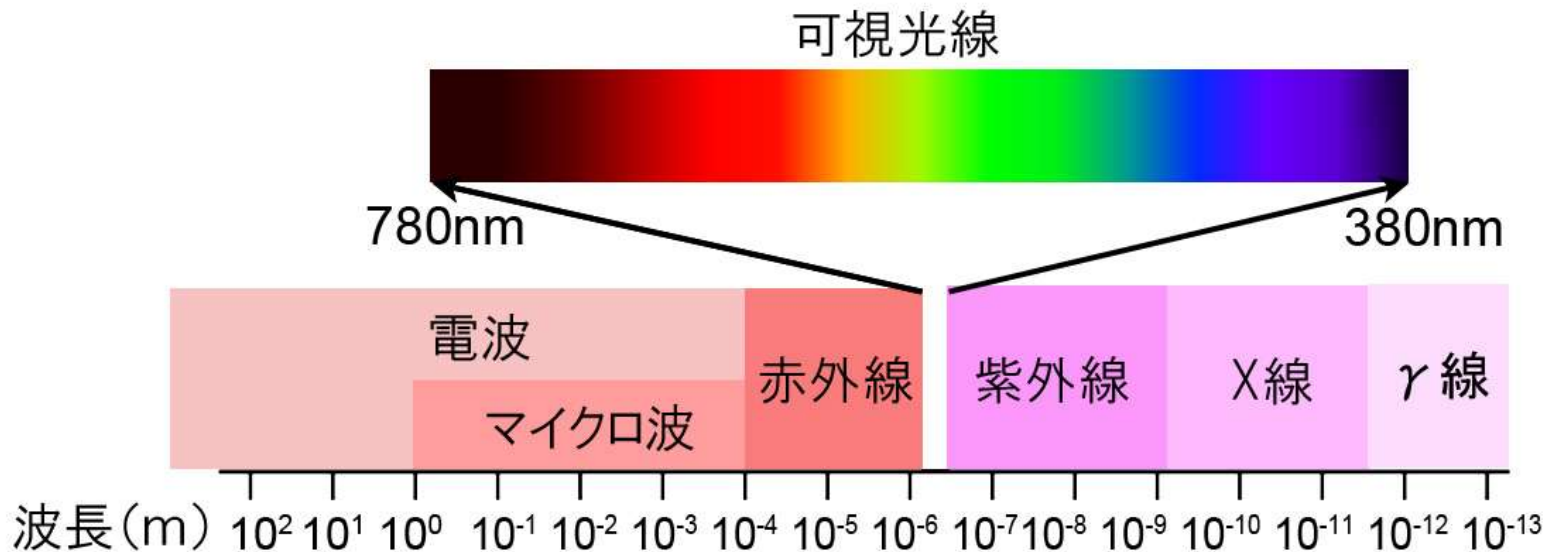


- そもそもどのように値を決定するのか？
 - デザイナーの感性と経験
 - 経験のあるカメラマンなら撮影するシチュエーションを見ればだいたいの露出値はわかるが...
 - ダイナミックに露出値が変化する時も手動で対応
 - いちいち場所ごとにScene EVを設定？
 - 大きな建物や木の陰など
 - そのためにコリジョン構造や階層構造を設定？

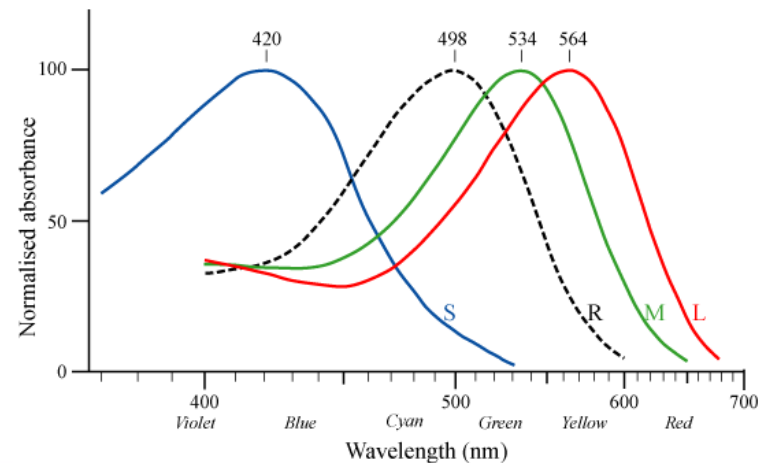


- 物理ライティングにすれば
 - そもそも露出を正しく計算できるのでScene EVのようなパラメータはいらない
 - デザイナーがいちいち設定する必要はない
 - 正確で物理的なトーンマップも可能
 - よりクオリティの高いトーンマップ
 - 自動的にシーンに合った露出になるので
 - 違和感のないカメラ演出が自動的に可能

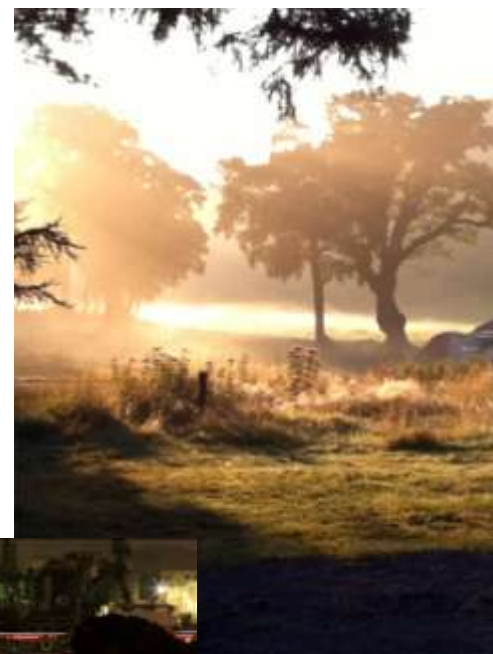
- 波長400nm～700nm程度の電磁波
 - 電磁波に起因する物理現象が各種光学現象や色にまつわる視覚的な効果を生み出す



- RGB表現とは波長軸上の連続信号(電磁波)を3つの基底で近似したもの
 - なぜ?
 - 人間の目が3原色だから
 - 3原色でもいろいろな基底(色空間)が存在する
 - sRGB, CIE-RGB, XYZ, Lab, YCbCr, xvYCC, Adobe RGB...



- 人間の目が3原色だからといってRGBでレンダリングしてよいのか？
 - 波長によって変化する物理現象が(シミュレーションで)正しく再現できない
 - 屈折散乱現象
 - コースティクス
 - 大気散乱
 - 煙や霧
 - 干渉現象
 - 薄膜干渉
 - 塗装
 - 光学現象
 - 回折(グレア)
 - 収差
 - 光源とBRDF



スペクトルレンダリングをRGBに変換したもの

$$Col_{R,G,B} = \int V_{R,G,B}(\lambda) \rho(\lambda) L(\lambda) d\lambda$$

~~≠~~ ?

$$Col_{R,G,B} = \int V_{R,G,B}(\lambda) \rho(\lambda) d\lambda \int V_{R,G,B}(\lambda) L(\lambda) d\lambda$$

通常のRGBレンダリング

$V_{R,G,B}$: 色空間における各RGBの基底

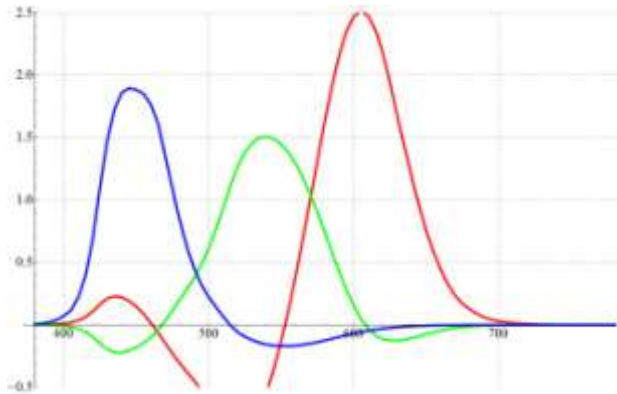
ρ : 波長のみを考慮したBRDF(反射率)

L : 入射光

- 仮想的なわかりやすい光源で比較

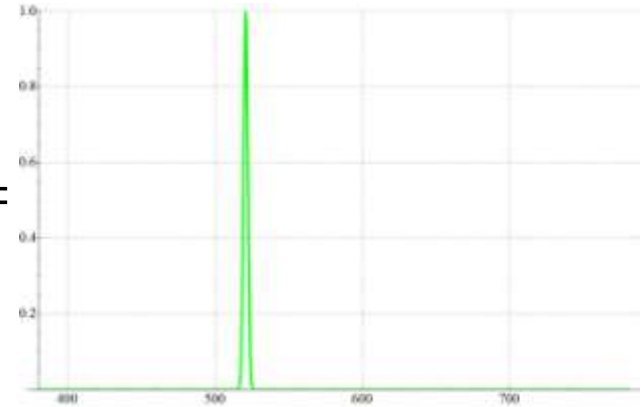
$$V_{R,G,B}(\lambda) =$$

各チャンネルの
等感度曲線



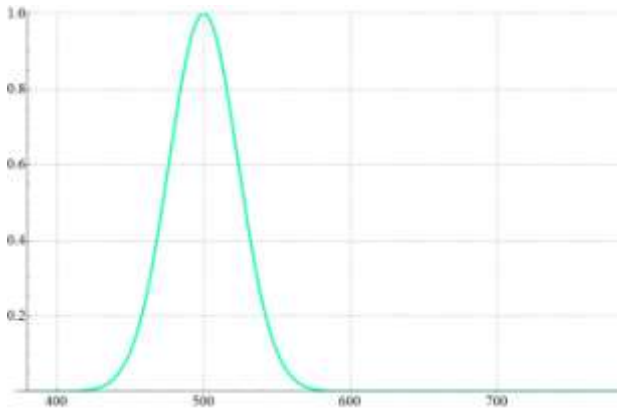
$$L_1(\lambda) =$$

ある光源1の
スペクトル



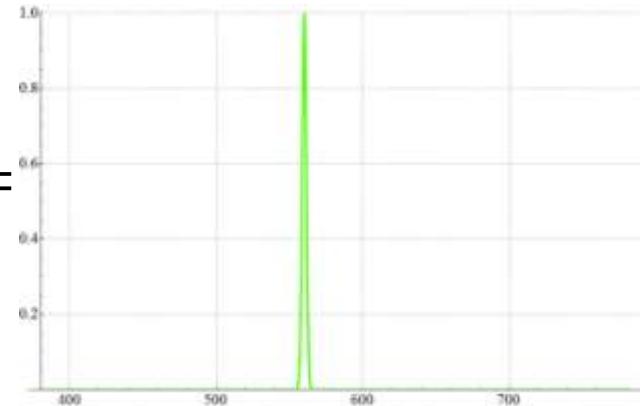
$$\rho(\lambda) =$$

あるBRDF
(反射率)



$$L_2(\lambda) =$$

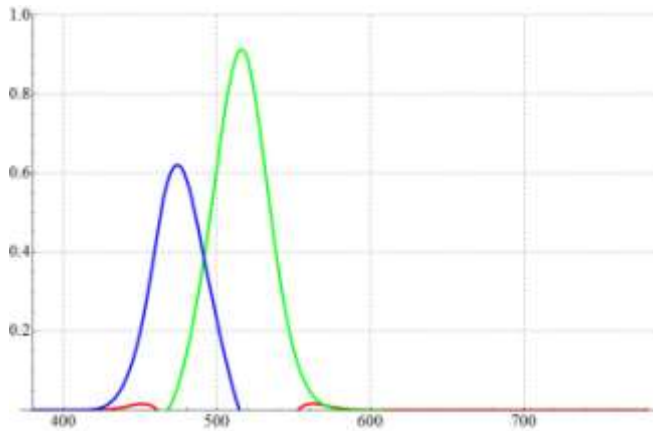
ある光源2の
スペクトル



RGBで計算すると?

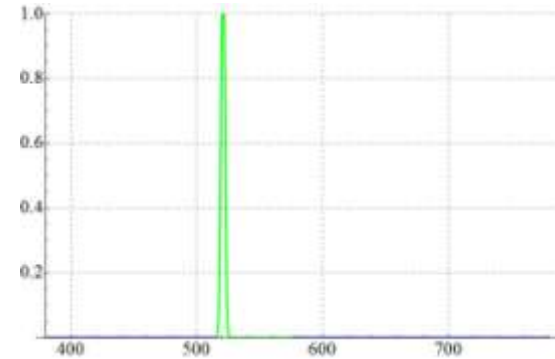
第1項はBRDF(反射率)であるから
光源によって変化しないので

$$\text{第1項} = k \int V_{R,G,B}(\lambda) \rho(\lambda) d\lambda = (-0.7836, 1.008, 0.6097)$$



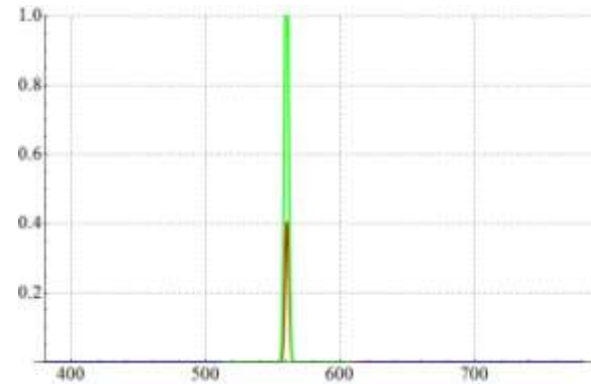
光源1の場合

$$\text{第2項} = g \int V_{R,G,B}(\lambda) L_1(\lambda) d\lambda = (-0.7130, 1.001, -0.049)$$



光源2の場合

$$\text{第2項} = g \int V_{R,G,B}(\lambda) L_2(\lambda) d\lambda = (0.3090, 1.001, -0.1285)$$

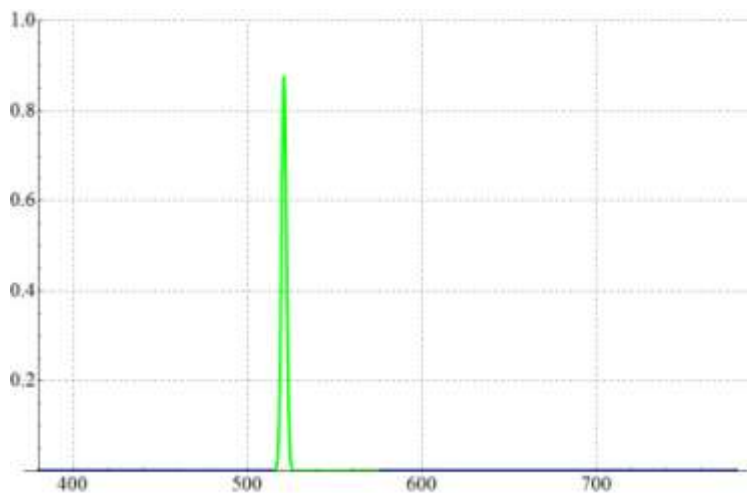


k, g は値を見やすくするための適当な正規化係数

光源1の場合

$$k \int V_{R,G,B}(\lambda) \rho(\lambda) L_1(\lambda) d\lambda$$

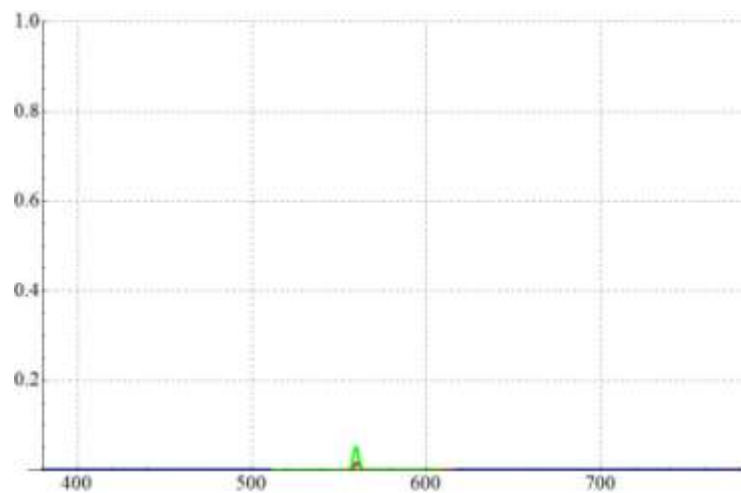
$$= (-0.7140, 1.0000, -0.0486)$$












光源2の場合

$$k \int V_{R,G,B}(\lambda) \rho(\lambda) L_2(\lambda) d\lambda$$

$$= (0.0175, 0.05864, -0.0075)$$



k は値を見やすくするための適当な正規化係数

反射率 	光源1 	光源2 
RGB計算 (クランプなし)	(0.5587, 1.009, -0.0298) 	(-0.2421, 1.009, -0.0783) 
RGB計算 (クランプあり)	(0.0, 1.009, 0.0) 	(0.0, 1.009, 0.0) 
スペクトル計算	(-0.7140, 1.0, -0.0486) 	(0.0175, 0.05864, -0.075) 

「クランプあり」とは

より現実的な計算を行った場合のこと

負の値はsRGBの色域外なので0でクランプして計算する

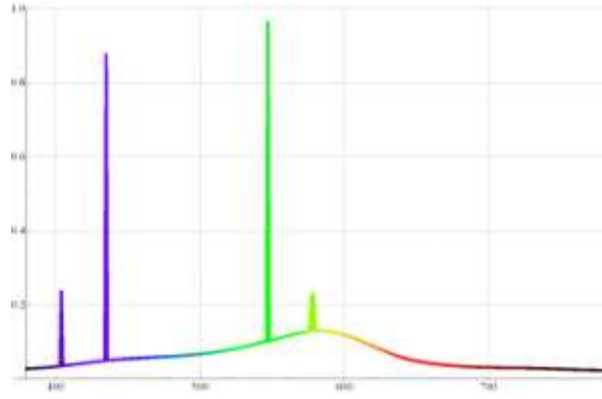
たとえばテクスチャに負の値を格納しないということ

- 検証に使った光源のスペクトルは現実的ではないのではないか？
 - 実際にはあのようなスペクトルは限定的
 - 低圧ナトリウムランプ
 - 一部の波長にピークが立っているスペクトルは人工光源においては珍しくない
 - 演色性
 - 平均演色評価数(Ra)

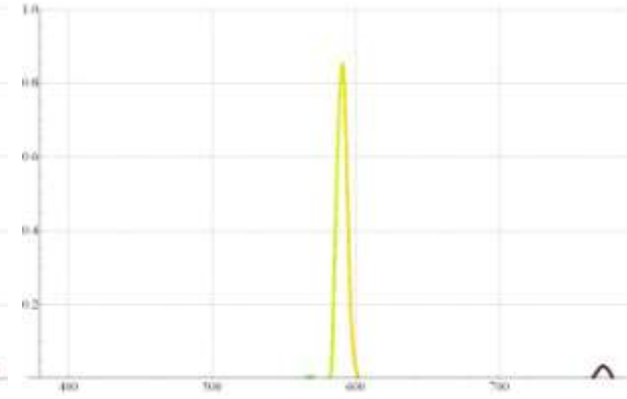
光源の例



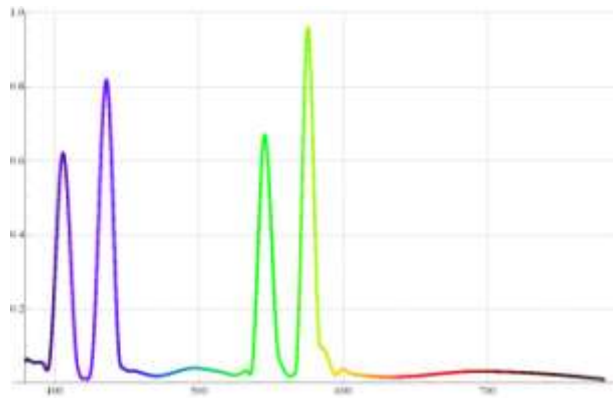
白熱電球



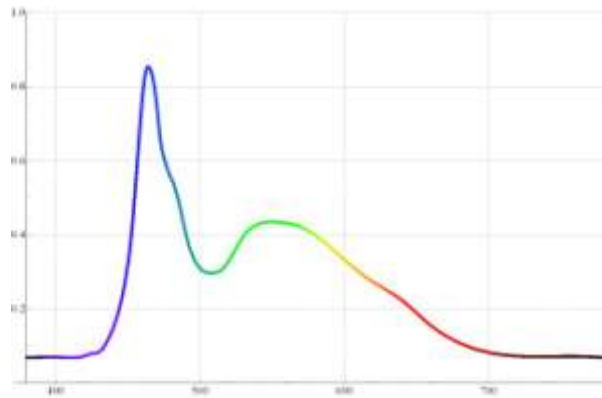
蛍光灯



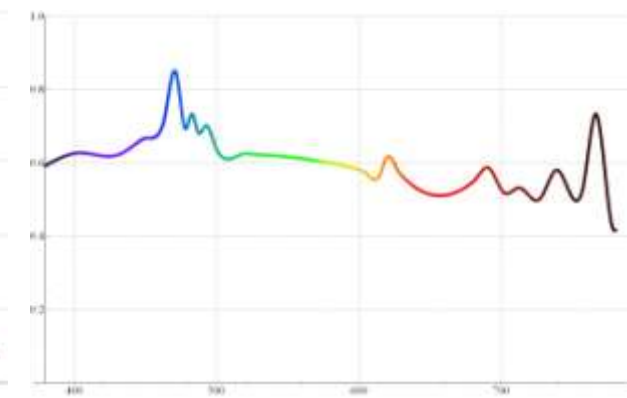
低圧ナトリウムランプ



高圧水銀ランプ



白色LED



キセノンランプ

- もしレンダリングが3原色のみで充分だと仮定するならば
 - ある色を2種類の光源で撮影してRGBに変換したものはホワイトバランス調整で全く同じ色にすることが可能はず
 - ホワイトバランス調整はR,G,Bのバランス調整なので光源の色のバランスを(RGB上で)調整したことと同じになる

$$Col_{R,G,B} = \underbrace{W_{R,G,B}}_{\text{ホワイトバランスの調整は単なるスケーラーなので}} \int V_{R,G,B}(\lambda) \rho(\lambda) d\lambda \int \underbrace{V_{R,G,B}(\lambda) L(\lambda) d\lambda}_{\text{光源の色を(RGB上で)調整したことと同じ}}$$

ホワイトバランスの調整は
単なるスケーラーなので

光源の色を(RGB上で)
調整したことと同じ

太陽光



D65光源



蛍光灯



UHP

太陽光



D65光源



蛍光灯



UHP

ホワイトバランス調整済み

D65光源



蛍光灯



タングステンランプ

UHP

ホワイトバランス調整済み

- RGBでは光の物理現象を考慮した色を正しくレンダリングできない
 - わかりやすい屈折や回折だけの問題ではない
 - 1つの物質(色)がさまざまな光源に照らされたときのレンダリングの結果すら正しく表現できない
 - ライトやBRDFのデータをRGBで持ってスペクトル化しても効果が薄い
 - テクスチャやライトをスペクトルでオーサリングする必要がある
 - 実装は簡単だが现阶段では現実的ではない
- スペクトルレンダリングでは物質がどんなライト環境下でも正しくライティングされる

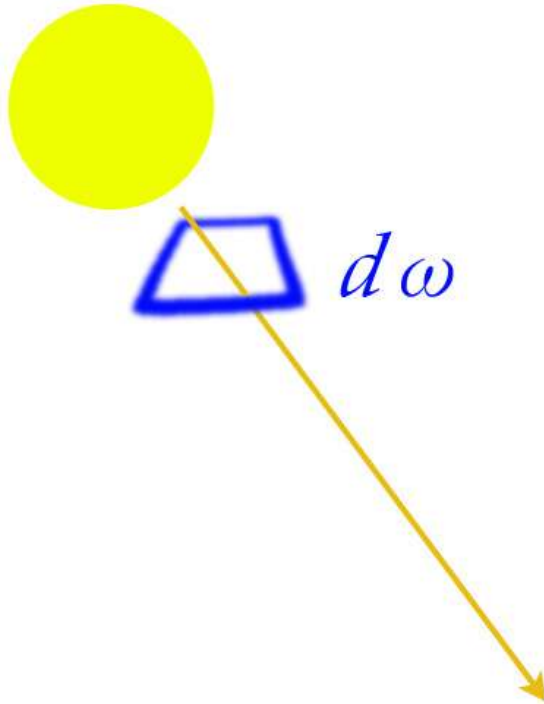


- 光(電磁波)のエネルギー量を表現するには
 - W(ワット)
 - 放射束(radiant flux)
 - もっとも基本的な単位
 - 純粹に光のもつエネルギー量を表す
 - 1Wの電気エネルギーがすべて電磁波に変換された場合の放射束は1Wのエネルギーを持つ

$$\Phi = \frac{dQ}{dt}$$

ただし $Q = \int_0^{\infty} n_{\lambda} \frac{hc}{\lambda} d\lambda$

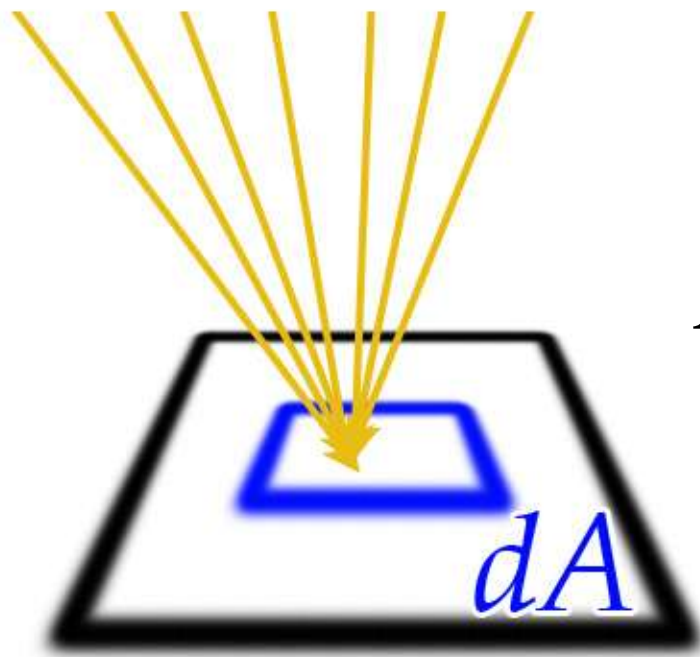
- 放射強度(radiant intensity)
 - W/sr
 - 1srあたりの光エネルギー量



$$I(\omega) = \frac{d\Phi}{d\omega}$$

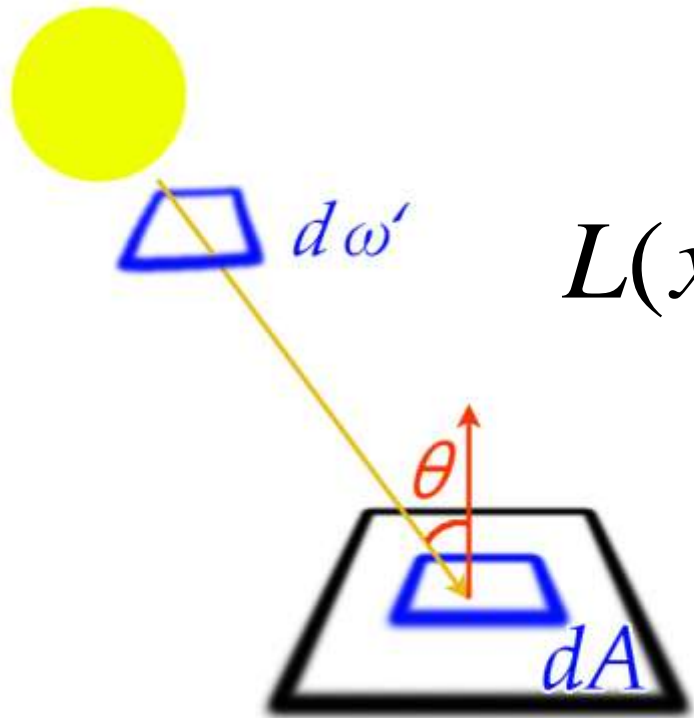
放射照度

- 放射照度(irradiance)
 - W/m²
 - 1m²辺りの光エネルギーの量



$$E(x) = \frac{d\Phi}{dA}$$

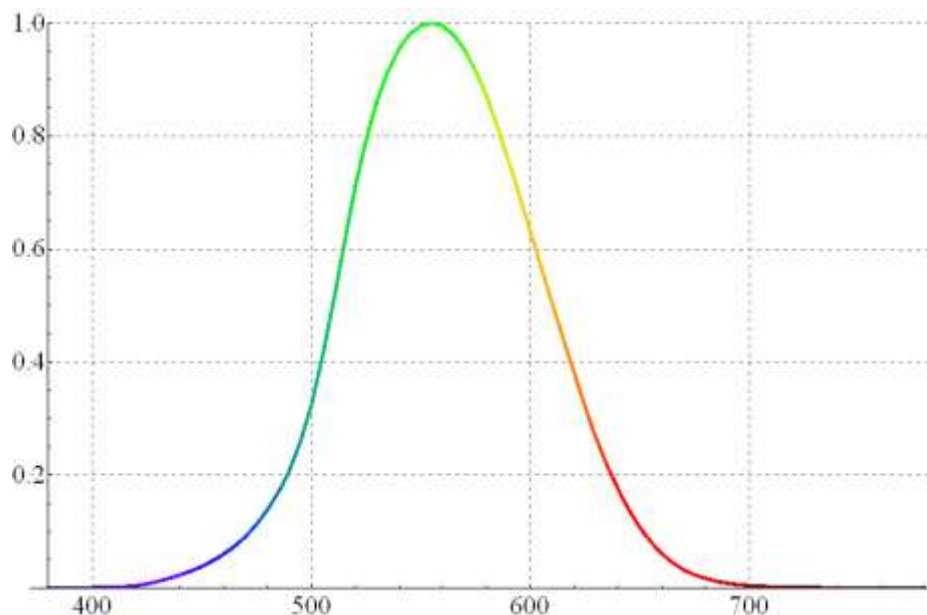
- 放射輝度(radiance)
 - W/sr/m²
 - 1sr、1m²辺りの光エネルギーの量



$$L(x, \omega') = \frac{d^2\Phi}{\cos\theta dA d\omega'}$$

- 人間の眼の性質を考慮した光の単位
 - 標準比視感度曲線による重み付けをする
 - 眼が各波長に対してどのくらい反応するかを表した関数

$$v(\lambda) =$$

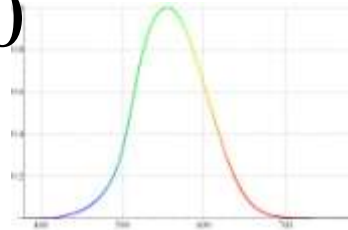


明所視標準比視感度曲線

- 光束(luminous flux)
 - lm(ルーメン)
 - 放射束(W)に対応する
 - 照明の光の強さを表すのに使用される

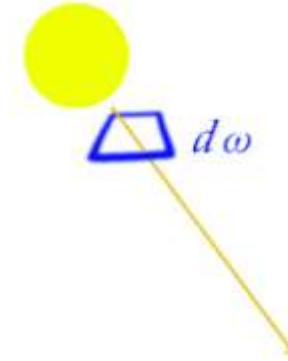
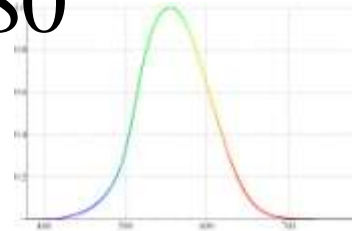
$$\Phi_v = K_m \int_{380}^{780} V(\lambda) \Phi(\lambda) d\lambda$$

$$K_m = 683.002$$

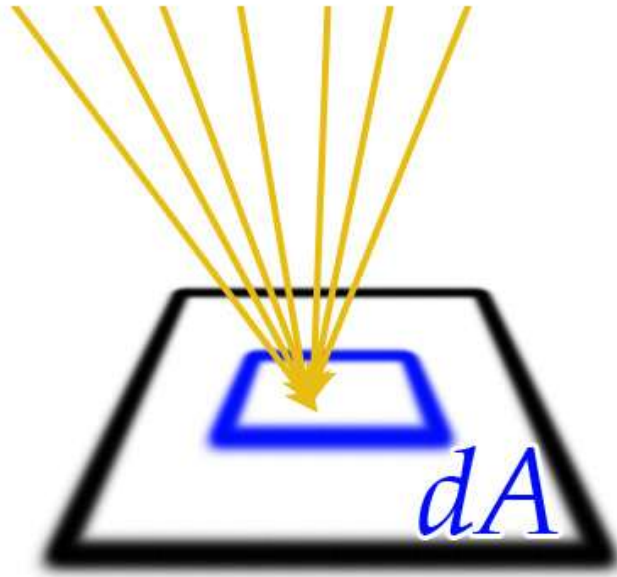


- 光度(luminous intensity)
 - cd(カンデラ)
 - 放射強度に対応する
 - 照明の光の強さを表すのに使用される

$$I_v = K_m \int_{380}^{780} V(\lambda) I(\lambda) d\lambda$$

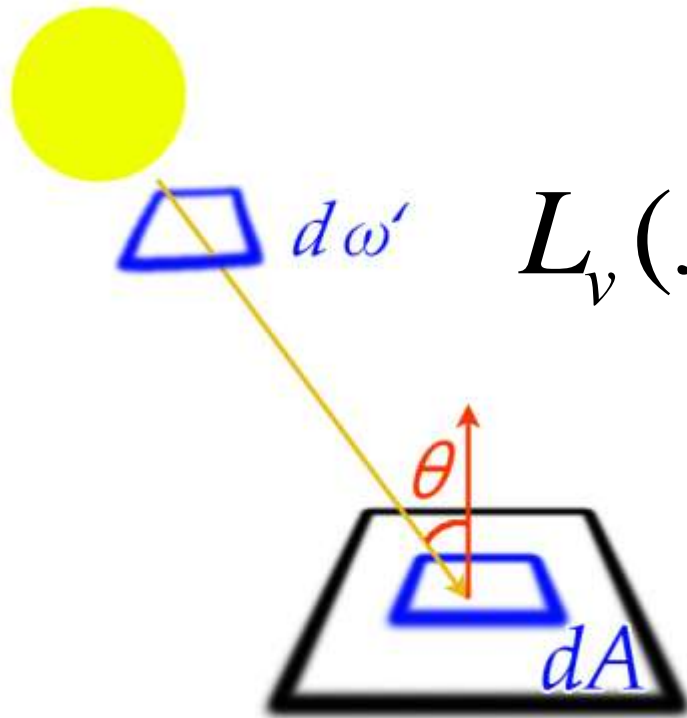


- 照度(illuminance)
 - lx(ルクス)
 - 放射照度に対応する
 - 照明の光の強さを表すのに使用される



$$E_v(x) = \frac{d\Phi_v}{dA}$$

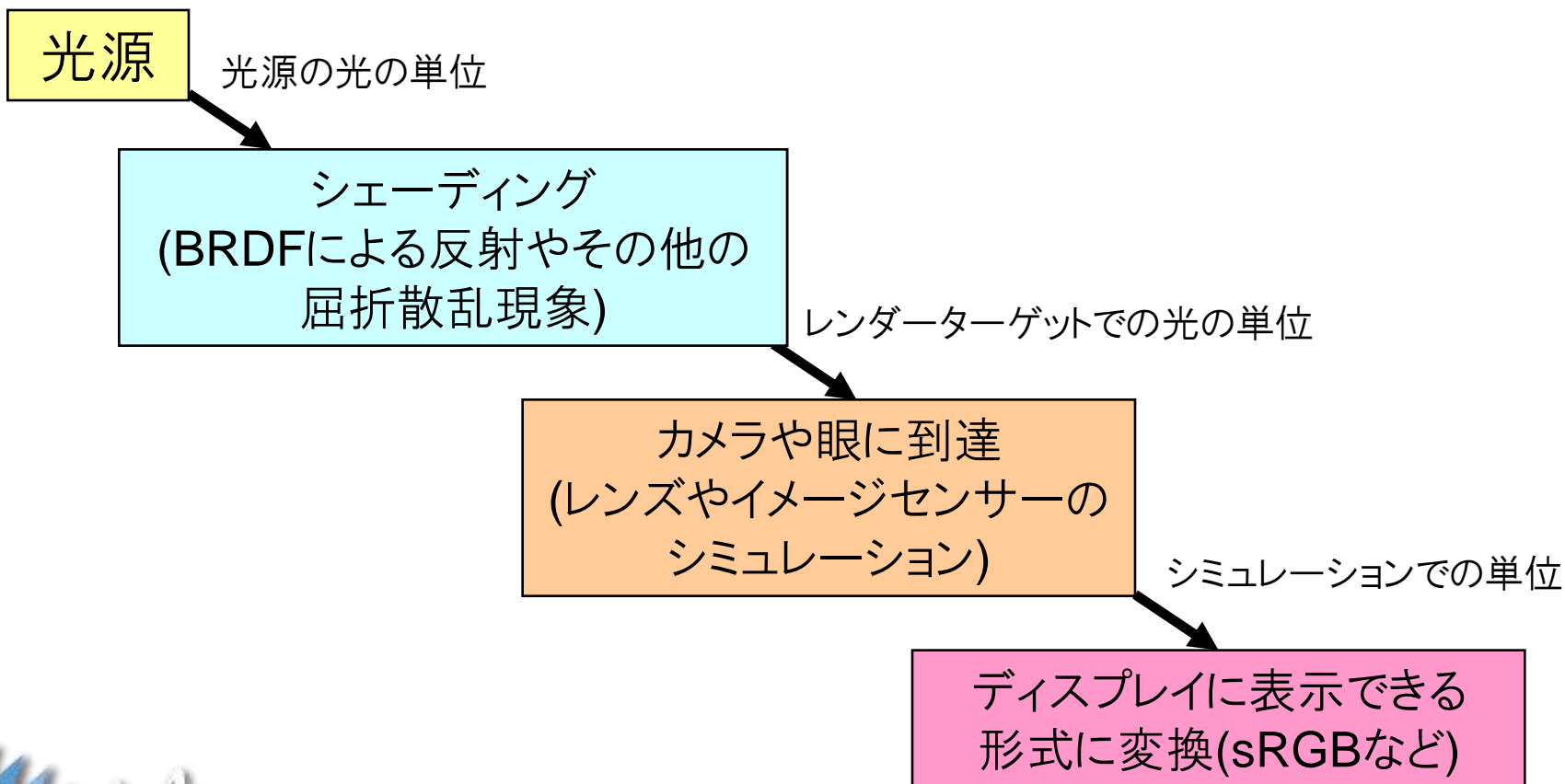
- 輝度(luminance)
 - cd/m²
 - 放射輝度に対応する



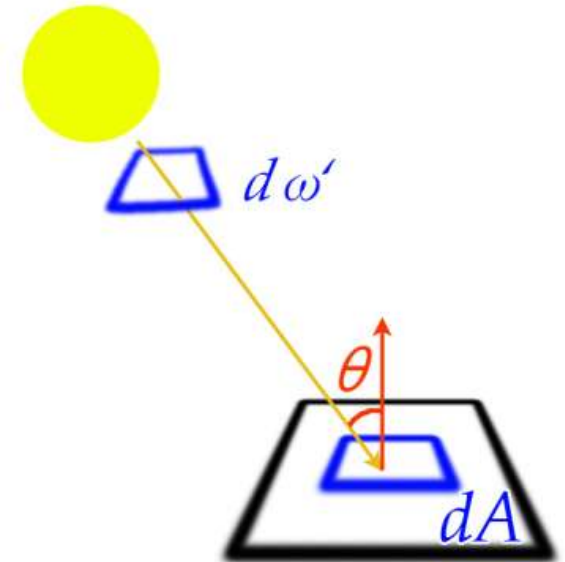
$$L_v(x, \omega') = \frac{d^2\Phi_v}{\cos\theta dA d\omega'}$$

	放射量	心理物理量
束	放射束(radiant flux) [W]	光束(luminous flux) [lm]
強度	放射強度(radiant intensity) [W/sr]	光度(luminous intensity) [cd]
輝度	放射輝度(radiance) [W/sr/m ²]	輝度(luminance) [cd/m ²]
照度	放射照度(irradiance) [W/m ²]	照度(illuminance) [lx]

- 光の表現を物理化するに当たって単位を設定したい

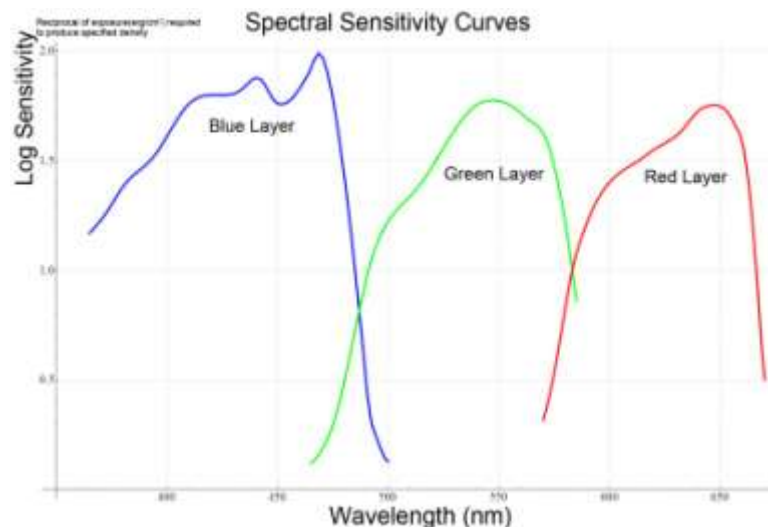


- 物理的に処理するなら放射束(W)?
 - レンダリングでは放射輝度に変換して
 - 放射照度をレンダーターゲットに出力?
 - しかしシェーディングで現時点では面積を正しく考慮していない
 - 放射輝度を考慮するのは無駄では?



欲しい物理量は何か？

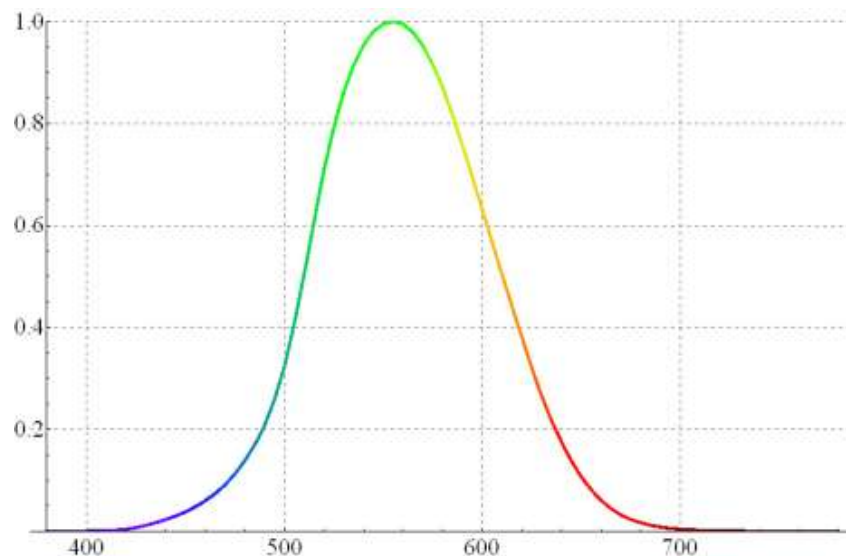
- 最終的に欲しい物理量を考える
 - ピクセルに格納されている光のエネルギー量が欲しい
 - 放射束？
 - フィルムやイメージセンサーの処理を考慮すると
 - データシートでは放射照度 × 露出時間(erg/cm²)を要求している



- 放射照度だと変換が少なくなる
 - 光源の単位として放射照度は物理的に不自然
 - 現時点では現実的な折衷案
 - 放射照度(光源)
 - シェーディング(露出量を考慮)
 - レンダーターゲット(露出×放射照度)

- 放射照度(W/m^2)を利用する
 - 設定する単位としてはわかりづらい
 - もうちょっとわかりやすい単位も必要
 - ルーメン(lm)、カンデラ(cd)、ルクス(lx)など

- 心理物理量の放射量への変換にはスペクトルが必要
 - スペクトルレンダリングをしている場合は定義を利用してそのまま変換可能
 - RGBの場合には近似が必要



- それっぽくなるように適当に近似
 - 464, 549, 612nmにおける比視感度をB,G,Rチャンネルの比視感度とし正規化する
 - 定義式に基づき光束(lm)を求める
 - 減衰カーブを適用し照度(lx)を求める
 - カラーから色温度を求める

$$lm = 683.002 \left(\frac{0.5069r + 0.911g + 0.044b}{1.462} \right)$$



- たとえば太陽光だと...
 - Wikipediaによると
 - 太陽光の日平均は32,000lxから100,000lx
 - シミュレータで計算すると
 - 2010年9月2日 東経139.635度 北緯35.46度
(パシフィコ横浜)

時間	放射照度 (W/m ²)	色温度(K)	照度(lx)
16:00	439.451	4490 (0.411,0.336,0.253)	107,944
17:00	347.288	3718 (0.459,0.339,0.202)	89,334
18:00	24.679	1340 (0.814,0.186,0.0)	6,710

- 照明のワット数は消費電力であり明るさそのものではない

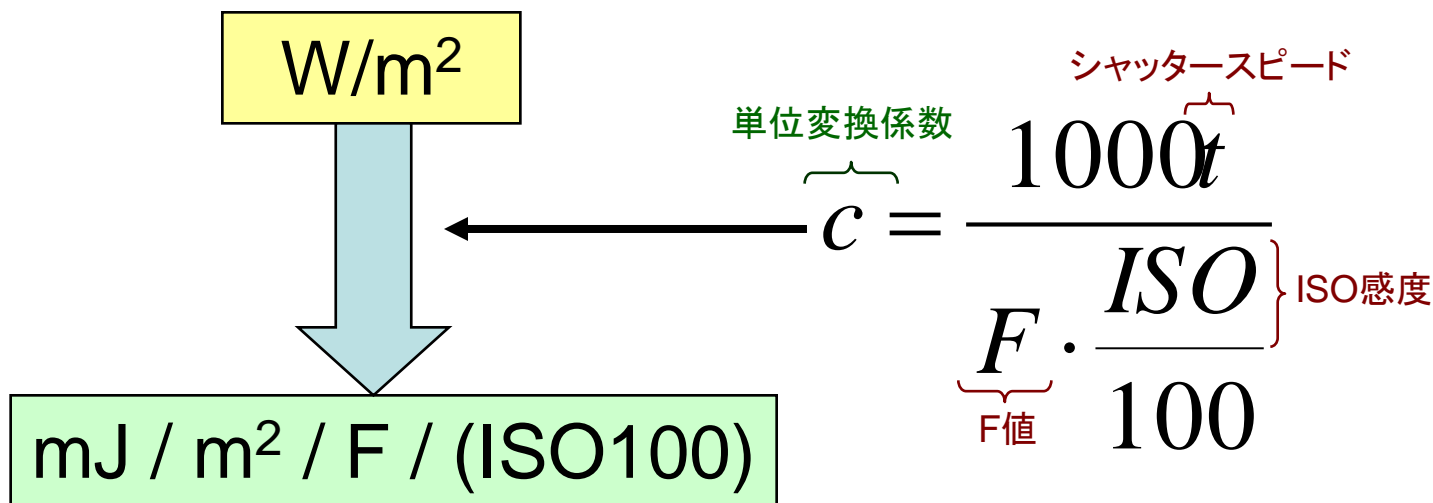
- 発光効率(lm/W)に注意すること

- 電力が電磁波に変換される率
- 電磁波のうち可視光線が占める率
- 心理量の場合は比視感度も考慮
- 太陽の場合(太陽定数:1,366W/m²)
 - 48%程度が可視光線
 - 大気で吸収散乱される分も考慮すると地上に届く光源としての太陽光は数十W/m²から500W/m²程度

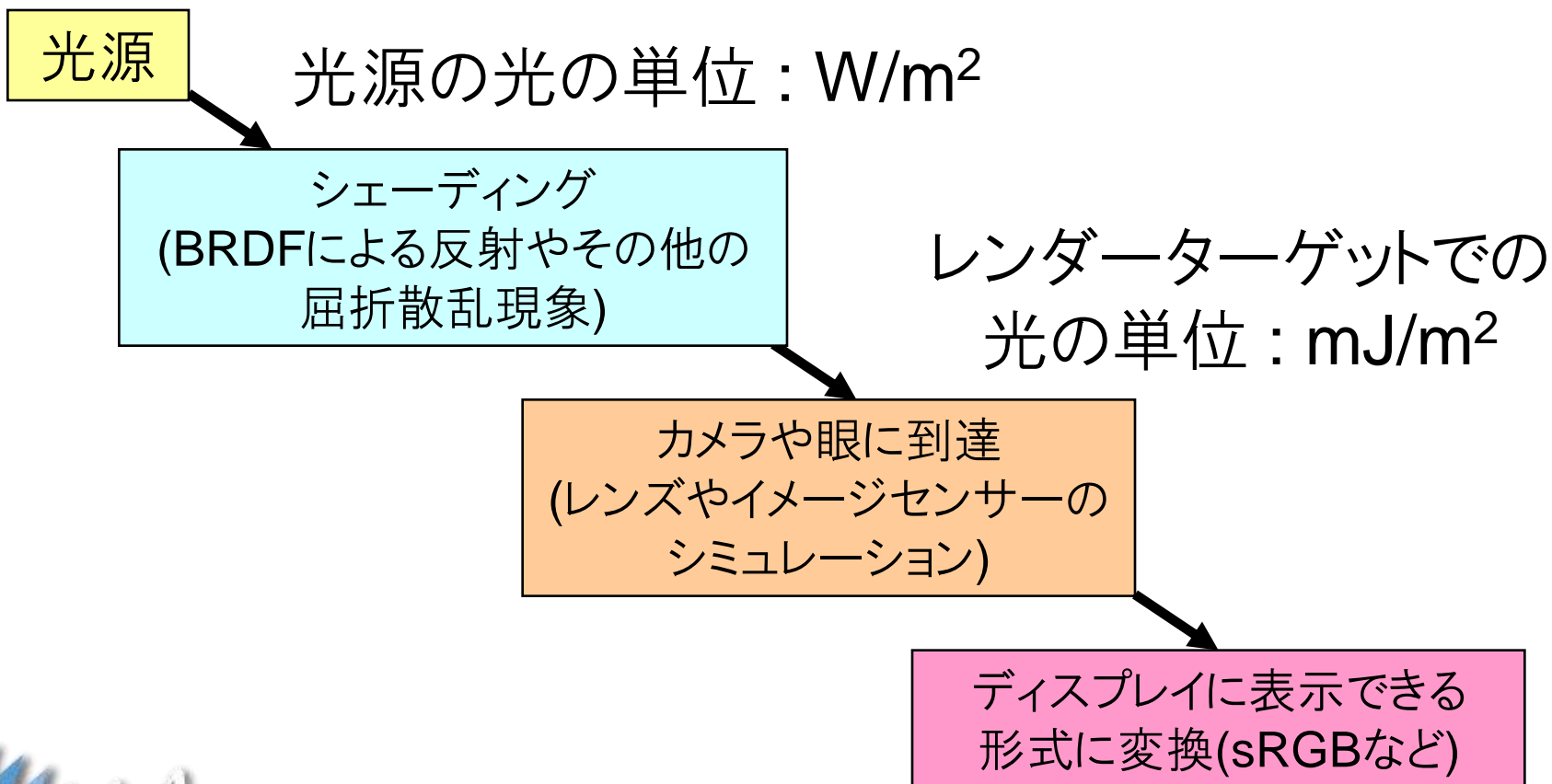


レンダーターゲットでの単位

- 光源から(物体で反射して)届いた光(W/m²)がレンズを通過してイメージセンサー(レンダーターゲット)に到達したと仮定
 - レンズパラメータを考慮して単位変換を行う



- 光を物理化するに当たって設定された単位

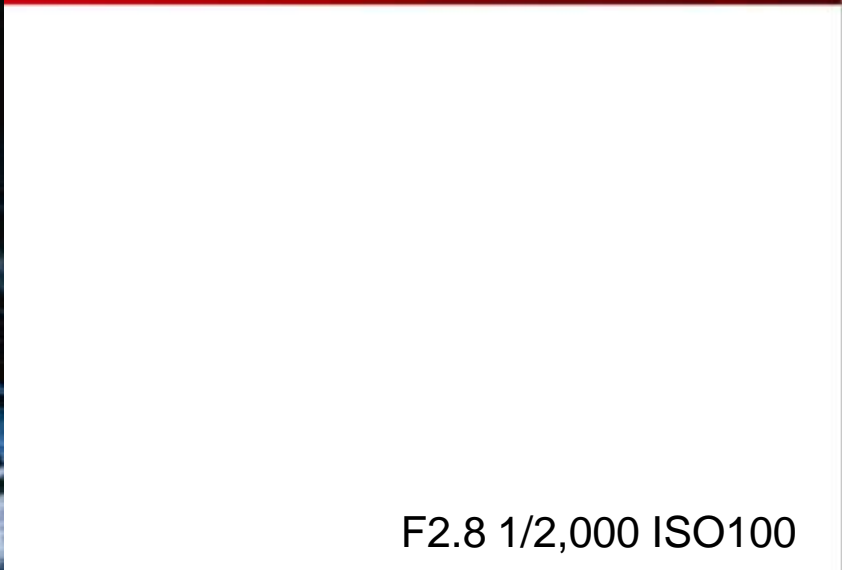


シーンの露出事例



F5.6 1/500 ISO100

キーライトの明るさ
およそ90,000lux



F2.8 1/2,000 ISO100



シーンの露出事例



F5.0 1/60 ISO100

キーライトの明るさ
およそ15,000lux

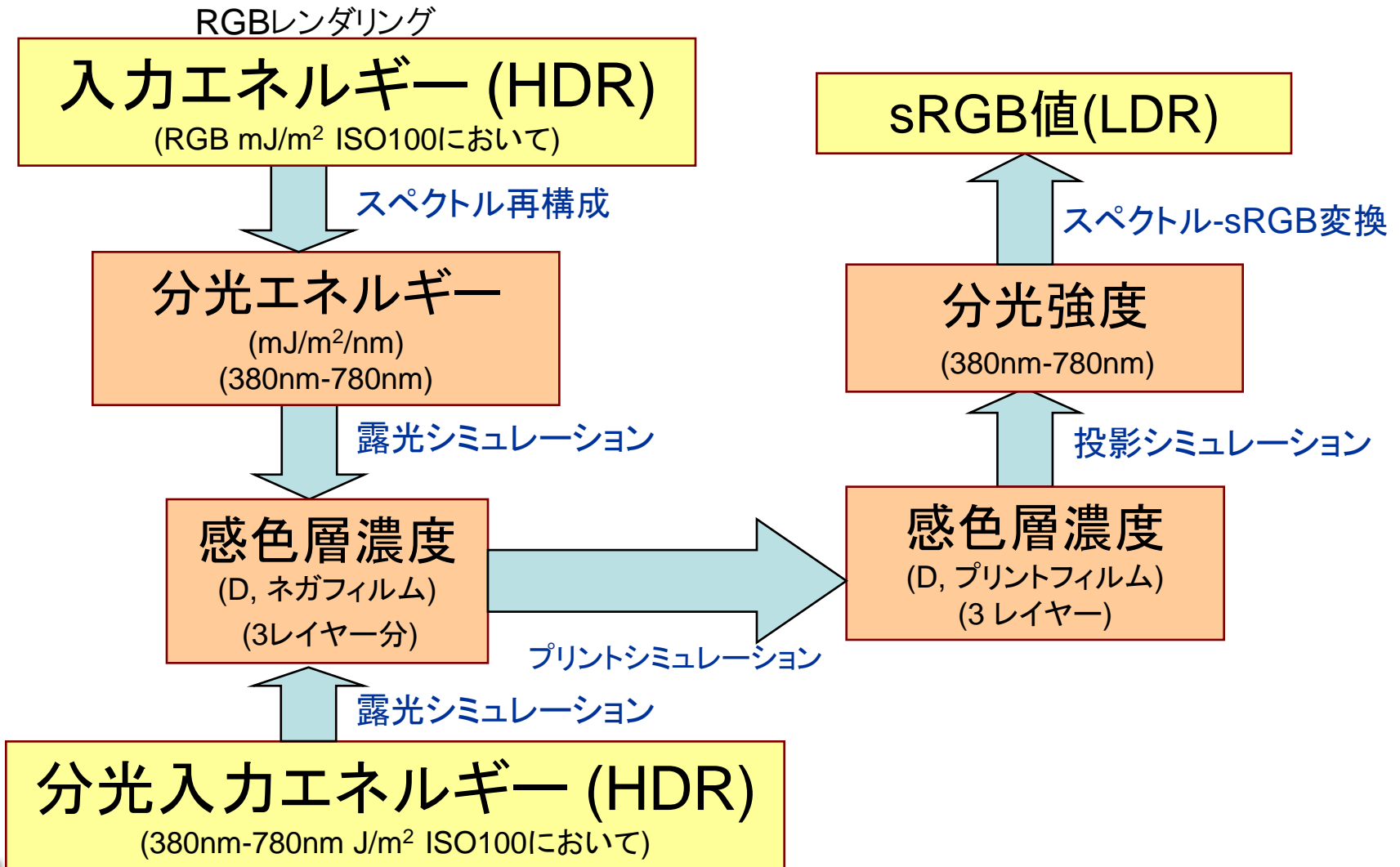


F2.8 1/200 ISO100

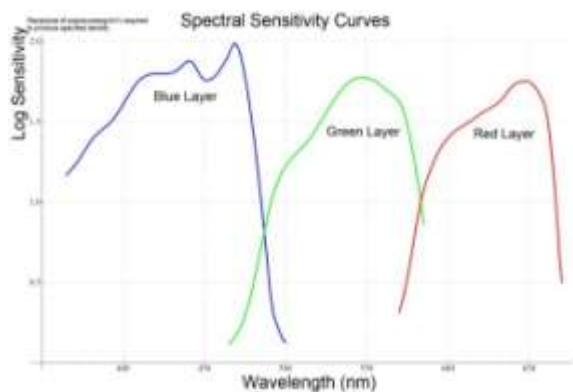
- 物理的に得られたレンダーターゲットにおける光エネルギーをsRGBに変換する
 - 適当に変換する
 - なんらかのデバイスを再現する
 - デジタルセンサー
 - アナログフィルム
 - 人間の眼



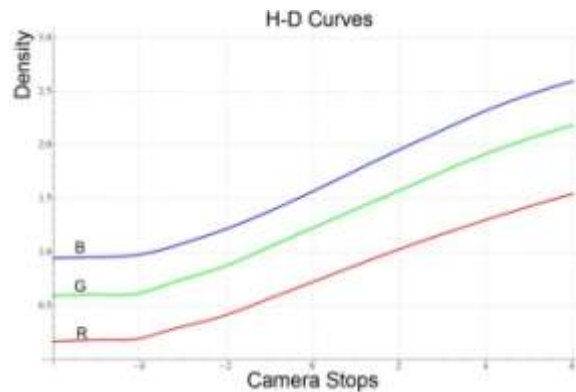
フィルム再現の概要



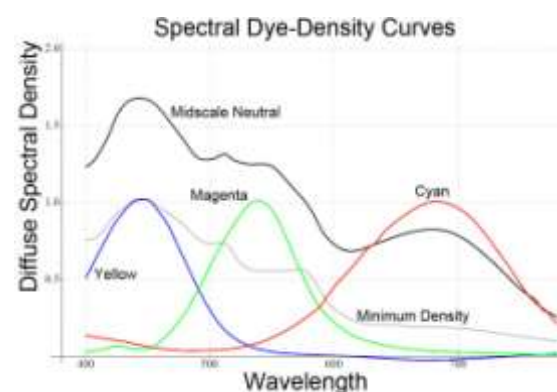
- フィルムのスペックシートに書かれているデータを利用する
 - 必要な情報がすべて含まれているわけではない
 - 足りない情報は補完する



分光感度曲線

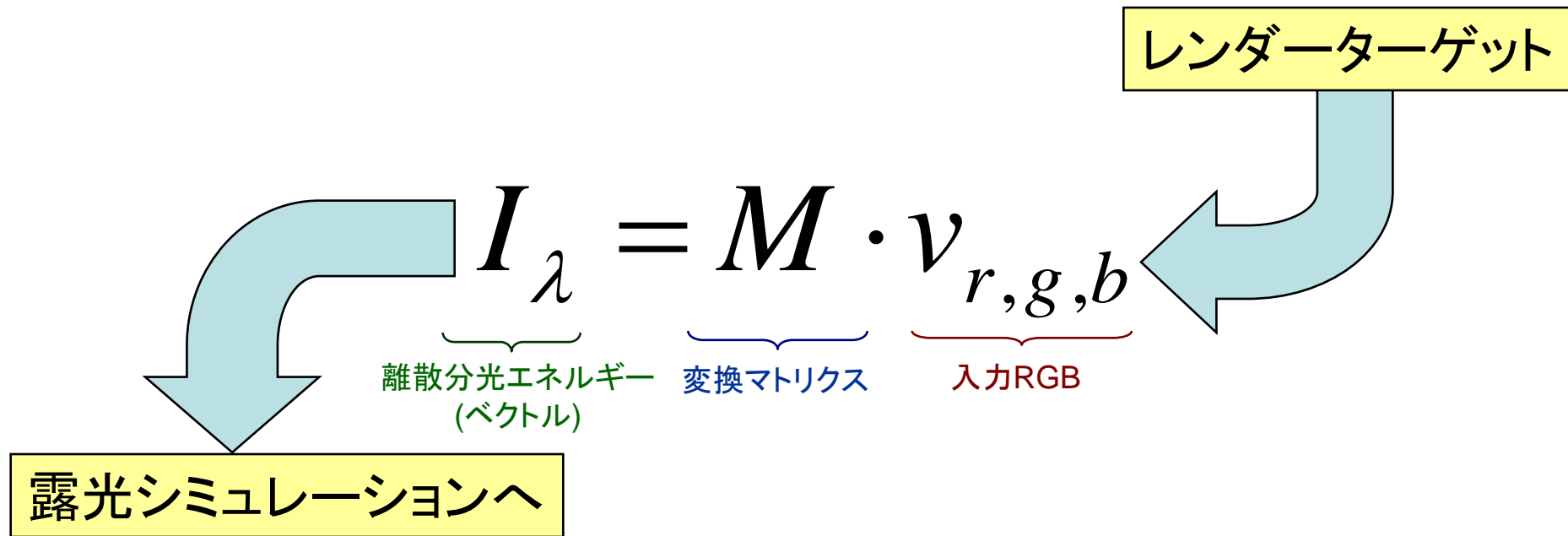


H-D曲線



分光濃度曲線

- RGBから分光エネルギーを得るために
スペクトルを再構成する
 - 変換マトリクスを利用する



- sRGBからスペクトル(A)、スペクトルからsRGB(B)の2つのマトリクスを同時設計する
 - スペクトルレンダリングも想定してマトリクスBは汎用的に設計する
 - XYZの等色関数を利用
 - マトリクスAに対して条件を設定しマトリクスを設計

- いくつかのマトリクス設計の条件を設定
 - 入力したカラーベクトルをスペクトル変換してそのままsRGBのカラーベクトルに変換した場合に
 - 入力(1,0,0)が変換後(1,0,0)に戻る
 - 入力(0,1,0)が変換後(0,1,0)に戻る
 - 入力(0,0,1)が変換後(0,0,1)に戻る
 - 入力(1,1,1)が変換後(1,1,1)に戻る
 - 任意の入力ベクトルと変換後のベクトルの大きさが等しくなるように
 - 可能な限りスペクトルカーブの連続性を保つ

マトリクスによる違い(ポジ)



標準マトリクス

実験的に作成したマトリクス



マトリクスによる違い(ネガ)



実験的に作成したマトリクス

標準マトリクス



- ネガフィルム(またはポジフィルム)の露光を再現して現像されたフィルムの各感色層の濃度を取得する
 - 露光シミュレーション方程式を使用

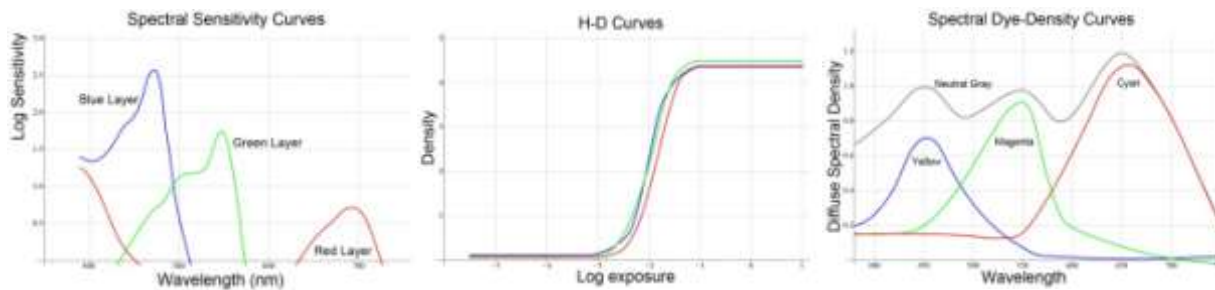
$$\bullet D_{r,g,b} = f_{r,g,b} \left(\log_{10} \left((\text{diag}(c_{r,g,b}) \cdot w_{r,g,b}) \cdot I_{\lambda} \right) \right)$$

- ネガフィルムを使用した場合にプリントフィルムにデュープする処理を再現する必要がある
 - ポジフィルムと異なりネガフィルムはそのまま観ても正しい色になっていない
 - あくまで撮影用に設計されている



- 実際には単純にプリントフィルムの特性を利用してネガフィルムのシミュレーションから出力されたスペクトルエネルギーをそのまま再撮影する
 - ネガフィルムを投影するライトが必要
 - プリントフィルムで指定されている光源とカラーフィルタを利用する
 - プリントシミュレーション方程式を利用

$$\sigma(\lambda) = I(\lambda) \cdot 10^{-\left(S_{\min}(\lambda) + c \cdot \sum_{r,g,b} \left(S_{r,g,b}(\lambda) \frac{D_{r,g,b} - D_{r,g,b}^{\min}}{D_{r,g,b}^{\text{mid}} - D_{r,g,b}^{\min}} \right) \right)}$$



- プリント処理にはキャリブレーションが必要
 - 正しいホワイトバランスと明るさを得るため
 - 18%グレイカードを適正露出で撮影したネガフィルムを現像する
 - このフィルムをプリントしプリントフィルムの仕様書で指定されている濃度を得られるように光源を調整する
 - 反復処理として実装
 - すでにカラーフィルタによる色分離が充分だったため実際には1回のループで正しいキャリブレーションが行われていた

2種類のキャリブレーション

- プリントフィルムの仕様書で示されている濃度になるようにスペクトルを調整
 - 分光濃度曲線の特性により18%グレーが完全なグレーにはならない
- グレーが等価中性濃度0.7になるようにスペクトルを調整
 - 18%グレーにおいて完全なグレーを得ることができる

キャリブレーションによる違い



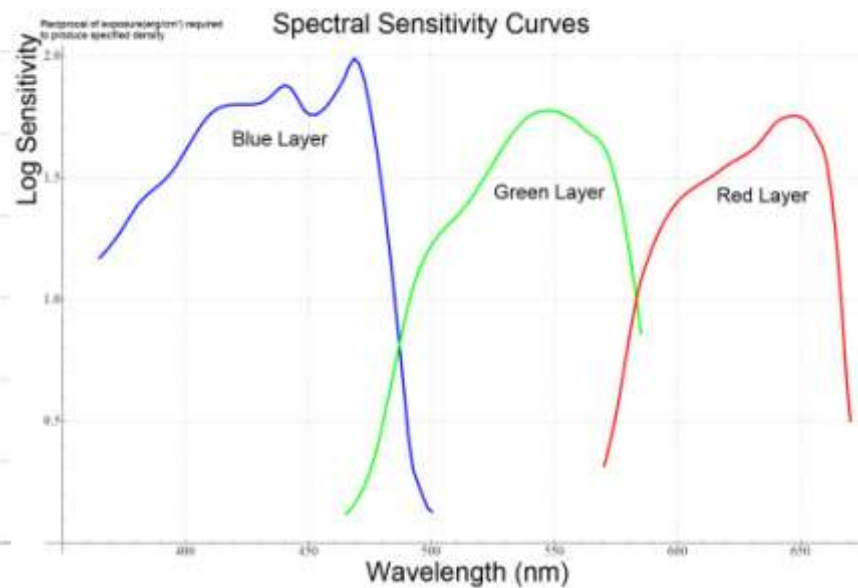
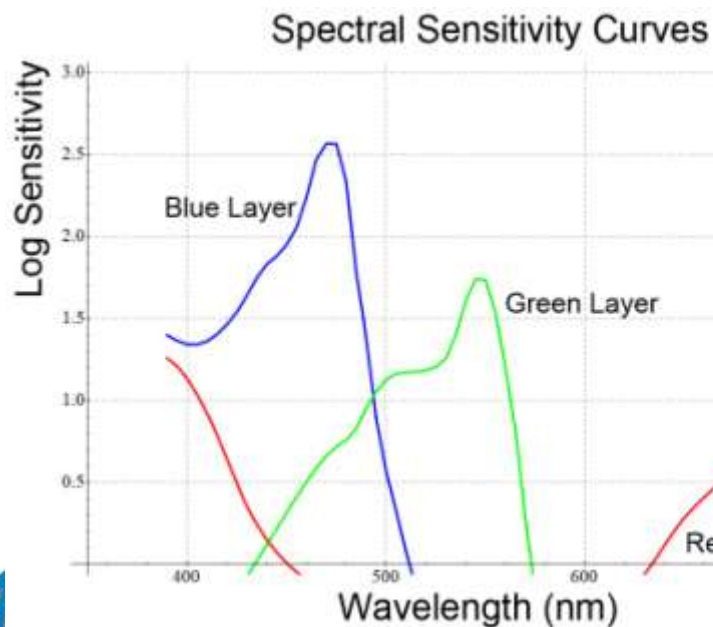
等価中性濃度0.7による
キャリブレーション

仕様書で指示されている濃度を利用したキャリブレーション

同じネガおよびプリントフィルムを利用したシミュレーション結果



- プリントフィルムの赤の感色層の感度のピークは撮影用フィルムに比べて赤外領域に近くなっている
 - おそらくカラーセパレーションのため
 - 結果的に赤の感色層が赤外線の影響を受けやすい
 - 800nm以上のスペクトルが結果に影響を与える



赤外領域による違い



赤外領域あり

同じネガおよびプリントフィルム
を利用したシミュレーション結果



赤外領域なし

- 最終的に現像されたフィルムを観るための投影シミュレーション
 - 分光濃度曲線が光源から届いた光をどのくらい透過するかを表している
 - 投影処理に利用する
 - 足りない情報に関しては補完する
 - 投影シミュレーション方程式を利用
 - $$\sigma(\lambda) = l_p(\lambda) \cdot 10^{-\left(\sum_{r,g,b} S_{r,g,b}(\lambda) \cdot D_{r,g,b} \right)}$$
 - 光源は6,500K or 5,500Kの理想黒体輻射スペクトルを利用
 - sRGB用(6,500K)
 - 映画館などでは通常キセノンランプが利用される(5,500K~6,000K)

投影ランプによる違い



5,500K

6,500K



ランプとフィルタの違い



5,500Kのランプで投影

6,500Kのランプで投影して
色温度フィルターで5,500Kにしたもの



- 最終的なスペクトルデータをsRGB色空間上のRGBデータに変換する
 - 変換マトリクスを利用して離散スペクトルベクトルをRGBベクトルに変換
 - この変換マトリクスはXYZ等色関数テーブルを利用
 - sRGB色空間のガンマ補正も同時に適用する

- このシミュレーション自体をリアルタイムにGPUで処理するには負荷が高すぎる
 - CPU上で計算しボリュームテクスチャに結果を書き込んでおく
 - 3D Look-Up Table
 - 32x32x32がパフォーマンス的には理想的なサイズ
 - 精度不足



- 対数領域に圧縮する
 - フィルムの持つISO感度はスケールファクタに変換して入力カラーに乗算する
 - U, V, W 座標を対数領域に変換する

$$U = 0.534577 + 0.217563 \cdot \log_2(u + 0.191406)$$

結果比較



Reinhard

F社ポジフィルム



結果比較



K社ポジフィルム

F社ポジフィルム



結果比較



F社ポジフィルム

K社ネガフィルム



旧手法との比較



旧手法によるK社ポジフィルム

今回の手法によるK社ポジフィルム



旧手法との比較



旧手法によるF社ポジフィルム

今回の手法によるF社ポジフィルム



旧手法との比較



旧手法によるK社ネガフィルム

今回の手法によるK社ネガフィルム



- ライトを物理化すると

- スペクトル

- 正しい光源、BRDF処理により正しい色が再現される
 - ライトや(BRDF)テクスチャもスペクトルで持つ必要がある
- 各種光学処理が正しく再現される

- エネルギー

- 実際の照明のデータを利用できる
- シミュレーション結果の物理量をそのまま利用できる
- 露出などが現実と同じように処理できる
 - Scene EVのようなハック的なパラメータをデザイナーが設定する必要がない
- 感度の物理的限界を正しくシミュレートできる

- ライトを物理化すると
 - 公開されているセンサーのデータを利用してその特性を再現できる
 - フィルムの持つ特性は以下のようなフィルタで単純には再現できない
 - カラー調整フィルタ
 - コントラスト / ブライトネスフィルタ
 - トーンカーブフィルタ
 - 彩度フィルタ
 - いろいろな(光源)環境下でも質感やライトを手動で調整することなく一貫した結果を得ることができる

- 研究開発部
 - 庄子達哉
- デザイナー
 - 金倉賢一, 重田一樹, 金子健一, 水上亮

<http://research.tri-ace.com>

フィルムシミュレーションについては上記サイトで
「Film Simulation for Video Games」のCourse Notes
で詳細に解説されています