

基礎からはじめる物理ベースレンダリング フォトリアル編

mebiusbox software

2019 年 2 月 17 日

目次

1	はじめに	2
2	フォトリアル	2
3	露出補正	3
4	光量	4
5	照明	10
5.1	照明の基礎	10
5.2	直接照明	13
5.3	間接照明	19
5.4	照明の単位	20
5.5	計測	20
6	色	20
6.1	色温度	21
6.2	色空間	21
6.3	XYZ と RGB の相互変換	24
6.4	色温度から RGB 色空間へ	24
6.5	標準の光源	25
7	カメラ	25
7.1	絞り, シャッタースピード, ISO 感度	26
7.2	露出値	26
7.3	露出補正	26
7.4	事前露出補正	28
8	さいごに	28
	参考文献	29

1. はじめに

物理ベースレンダリングといえば、簡潔に言うと物理的なパラメータ（メタルネスやラフネスなど）から物体の BSDF を計算し、レンダリング方程式を解くことです。これは光が物体に入射したとき、その光の量がある方向にどれだけ反射されるかを計算します。物体に入射してくる光は基本的に光源から発生し、光源には平行光源や点光源、面光源、スポットライトなどがあります。ここで、光源から発生する光の量を考えてみましょう。CG における光の物理量として最小単位は「光子（フォトン）」です。ある領域に単位時間当たりには通過する光子の集まりを「光束」といいます。レンダリングするときに計算する光の量としてはこの光束を扱います。ですから、光源から光束がどれくらい放射されるか指定することになります。例えば、現実にある太陽光やオフィス内の蛍光灯、電球と同じ光束にしたい場合はどうしたらよいでしょうか。これは扱っている CG ソフトやレンダリングエンジンによって変わってきます。そこで、光源と光束に関する基本的な知識を得られれば、一貫性の取れたライティング、より現実的、もしくは意図した描写を行うための適切な設定をすることが可能になると思います。ここでは光源と光束の基本的な内容を解説していきます。また、太陽光など実際に計測された値を使用すると、値が大きすぎて画面が真っ白になってしまう場合があります。光源の値を調整すれば対応できますが、現実と同じ値を設定しておくことで保守性も高くなりますし、なによりわかりやすいです。そのため、光源は変更しないで補正をすることになります。そこで導入するのが現実のカメラと同じパラメータを使った露出補正です。今回は絞り値やシャッタースピードといったカメラの内容にも触れ、最後にこれらをどのようにレンダリングに使用するかを解説していきます。

光の物理量を扱う場合、光学では放射量として扱いますが、今回では現実の光源やカメラを使います。それらのパラメータは人間の眼を通した光、つまり測光に基づいていますので、光の物理量を測光量として扱います。

2. フォトリアル

フォトリアルとは、「カメラ的に正しいフォトリアルグラフィック制作ワークフロー」によれば

写真（フォト）を忠実（リアル）に再現すること

と定義しています。写真はカメラを使って被写体を撮影して作成します。カメラ（デジタルカメラ）は簡潔にいうとレンズを使って集めた光を、撮像素子（イメージセンサー）が電気信号に変換し、プロセッサが画像データとしてメモリーカードなどに保存しています。それを忠実に再現するということは、このカメラと同じことをすればよいことになります。

それではカメラが写真を作成するために必要なものは何でしょうか。それは**光**と**被写体**です。光が被写体や他の物体に当たって反射し、カメラのレンズに集まったものが写真となります。この関係を式にしてみると


$$\text{光} \times \text{被写体} \times \text{カメラ} = \text{写真}$$

Fig.1: 光・被写体・カメラ

という感じになります。また、光の量が2倍になれば被写体の明るさも2倍になり、カメラが受け取る光の量が倍になれば、写真の明るさも倍になります。この関係は



Fig.2: 光（光量が2倍）・被写体・カメラ

となります。これをCGに当てはめると



Fig.3: 光源・被写体・CGカメラ

と考えることができます。ここで、物理ベースレンダリングで出てくるBSDF(拡散BRDF, 鏡面BRDF, BTDFなど合わせたもの)といったものは**マテリアル**に含まれています。BSDFは簡潔に言うと、入射した光がある方向に出射する光の割合です。先ほどの式を振り返ってみると光源から放射される光が、物体のマテリアルによってCGカメラの方向に反射する光の量が決定し、CGカメラがそれを受け取ってレンダリング画像となります。ここでCGカメラに注目してみると、CGカメラも物体によって反射された光に対して掛け算を行っています。一体何を掛けているのでしょうか。次はそれを見ていきましょう。

3. 露出補正

先程のところでも説明したように、カメラはレンズで集まった光を撮像素子が電気信号に変換します。このとき、レンズで集まった光を撮像素子に当てることを**露出**といいます。現実のシーンではカメラで撮影するとき、状況によって被写体の明るさが変わります。例えば、日中（晴天）の場合や曇っているとき、屋内や屋外といったように様々な状況が考えられます。どのような状況でも同じ明るさの写真を撮るためには、この露出を調整する必要があります。これを**露出補正**といいます。露出補正するために、カメラには**絞り値**、**シャッター速度**、**ISO感度**というものがあります。詳しくは後ほど取り上げますが、ここで簡単に説明しておきます。絞りは円環状の遮へい版で、レンズを通る光束を調整します。絞り値はこの絞りの開口径の値で、この値が小さいほど光束が多くなり、絞り値が大きいほど光束が少なくなります。次に、シャッター速度は、撮像素子の前にあるシャッターを開く時間です。シャッターが開いている時間だけ、撮像素子に光が当たり電気信号に変換されます。露出調整は基本として絞り値とシャッター速度を使います。調整された光束が撮像素子などに作用することを**露光**といいます。ここまでの少し整理しておきましょう。露出を次のように考えてみます。今、コップを手を持っていて、目の前に水道の蛇口があります。そこからは水が出ます。もしかすると温かいかもしれないし、水ではなくビールかもしれません。この水の量を光の量とってください。蛇口を開ければコップに水が溜まっていきますよね。この蛇口を少しだけ開けた状態だと水の出る量は少ないですし、全開にすれば多くの量が出てきます。この蛇口の開け具合が絞り値です。そして、蛇口を開けている時間が長いほどコップには水が多く溜まってきますよね。この開けている時間がシャッター速度です。そして蛇口の開け具合と開けている時間を調整してコップに入れる水の量を調整することが露出補正です。このコップに入れる水の量が常に同じであれば同じ明るさとなるわけです。

ですが、ちょっと待ってください。状況や場所によって水道の蛇口から出る水の量がそもそも違うかもしれません。自宅で台所にある蛇口と、お風呂場の蛇口、洗面所の蛇口のどれもが同じ開け具合で同じ水の量が出てくるとは限りません。これは被写体の明るさ、つまり光源の強さが状況によって変わるということです。蛇口から出てくる基本的な水の量（＝被写体の明るさ）が小さければ、絞り値やシャッター速度を調整しても全然水の量が足りないということもあるわけです。そこで **ISO 感度** が出てきます。これは基本的な水の量を増幅させるものです。カメラに話を戻すと、撮像素子が電気信号に変換するときに増幅します。このとき小さい値を無理やり増幅しているため、増幅率が高いとノイズが発生してしまうという問題があります。絞り値やシャッター速度についても設定できる値の範囲がある程度決まっているため、この絞り値、シャッター速度、ISO 感度の3つの値をうまく設定する必要があります。カメラには自動で調整してくれる機能が備わっているので、実際に写真を撮るときはカメラにお任せでもいいのですが、レンダリングに導入するためにはマニュアルで調整できるようにしなければなりません。少し長くなってしまいましたが、前のところで疑問だったCGカメラが物体によって反射された光に対して掛け合わせているものというのは、露出補正になります。

4. 光量

ここからは光についてみていきます。まずは光の量からです。冒頭で述べたように光の物理量の最小単位は**光子（フォトン）**です。ある領域（面）に単位時間当たりには通過する光子の集まりを**光束**といい、単位は**ルーメン (lm)** です。

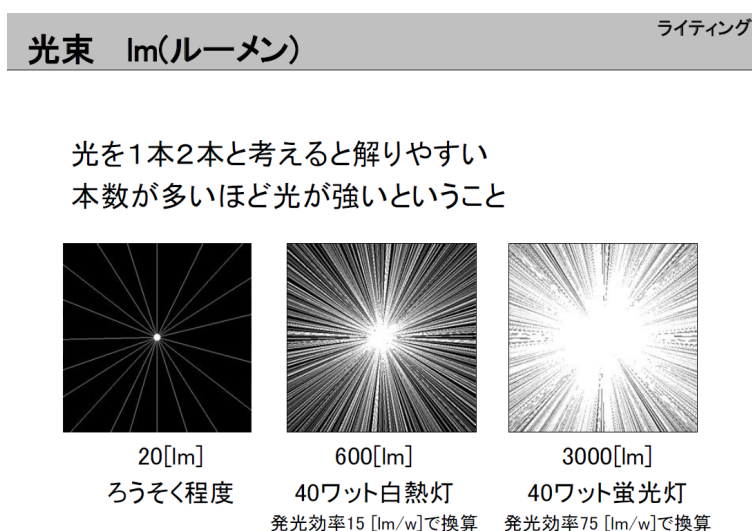


Fig.4: 鈴木雅幸「リニア空間と物理的に正しいライティング」株式会社キャビア

次に、点光源を考えてみましょう。点光源の中心位置から全方位に対して光が放射されます。このとき、ある方向への単位立体角当たりの光束を**光度**といいます。単位は**カンデラ (cd)** です。

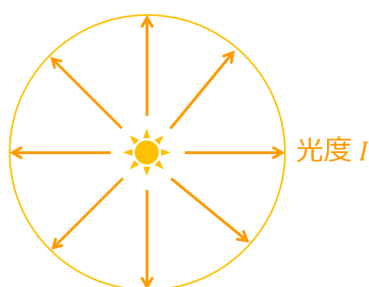


Fig.5: 光度

立体角とは空間での角度の大きさを、単位が**ステラジアン (sr)** です。これは半径 r の球を円錐状に切り取ったときに、切り取られた球面の面積が r^2 となる頂点の角です。

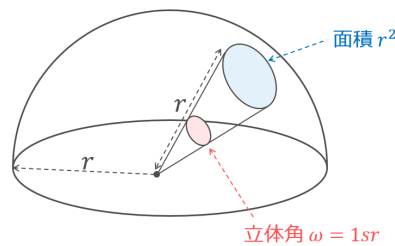


Fig.6: 立体角

ここで切り取られた球面の面積を A とすれば

$$\omega = \frac{A}{r^2}$$

という関係になります。このとき半径 $r = 1$ の単位球を考えると

$$\omega = \frac{A}{1^2} = A$$

となって、立体角は切り取られた球面の面積と一致します。つまり、立体角は球面の面積と考えることができます。球の表面積は $4\pi r^2$ ですので

$$\omega = \frac{A}{r^2} = \frac{4\pi r^2}{r^2} = 4\pi$$

という式が成り立ち、球の全立体角は 4π ということになります。立体角の単位を使えば、光度の単位は**ルーメン毎ステラジアン (lm/sr)** になります。

ある単位面積当たりに入射する光束を**照度**といいます。単位は**ルクス (lux, lx)** または**ルーメン毎平方メートル (lm/m²)** です。

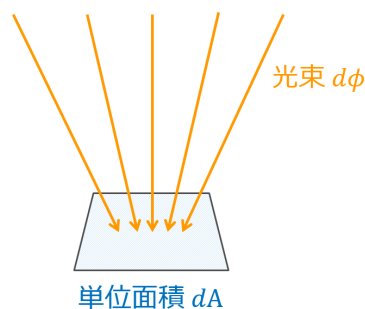


Fig.7: 照度

例えば、半径 r の球の中心に全方向に対して光度 I を放射する点光源を置いた場合、この点光源の全光束は $4\pi I$ となります。球の内側の総面積は $4\pi r^2$ なので、球面上の任意の点での照度は

$$\frac{4\pi I}{4\pi r^2} = \frac{I}{r^2}$$

となります。これは、照度が点光源の光度に比例し、距離の2乗に逆比例することを表しています。これを**逆2乗の法則**といいます。また、単位面積の面に対して垂直に光束が入射するとき

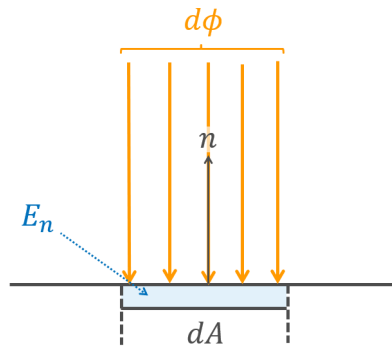


Fig.8: 単位面積に垂直に光束が入射

面の照度 E_n は

$$E_n = \frac{d\Phi}{dA}$$

です。ここで面が傾いていたときに

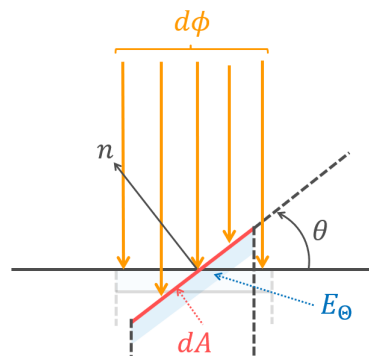


Fig.9: 入射角余弦の法則

傾いた面の照度 E_θ は

$$E_\theta = \frac{d\Phi \cos \theta}{dA} = E_n \cos \theta$$

となります。これは、ある領域の照度は光の入射角の余弦に比例するということです。これを**入射角余弦の法則**といいます。少し整理すると、点光源から 1cd の光度が放出されている場合、点光源が放射する光束は $4\pi \text{ lm}$ です。

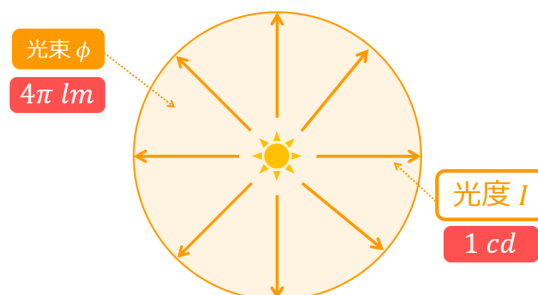


Fig.10: 光束と光度の関係

1cd は「 $540 \times 10^{12} \text{ Hz}$ (約 555nm に等しく、 $V(\lambda) = 1.0$ の波長に相当) の $1/683$ ワット (W) の単色光の与えられた光度」と 1979

年に定められました。つまり、1Wは683lmです。また、 $1m^2$ の面積がある領域に入射する光束が1lmのとき、その照度は1lxになります。

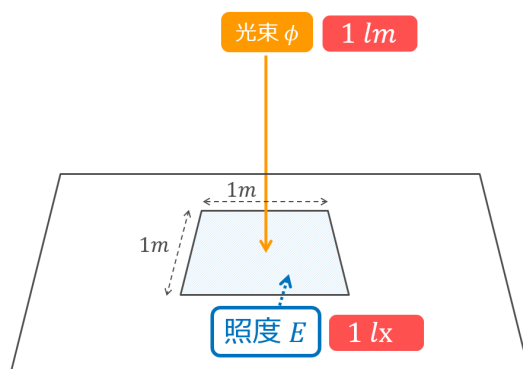


Fig.11: 光束と照度

例えば、60Wの白熱電球を考えます。白熱電球の効率を $10lm/W$ とした場合、この電球からは $60 \times 10 = 600lm$ の光束 Φ が放射されます。これを点光源とみなすと、この電球から放射される光度 I は

$$I = \frac{\Phi}{4\pi} = \frac{600}{4\pi} = 48lm/sr$$

となります。この電球から50cm離れたところの照度 E は

$$E = \frac{I}{r^2} = \frac{48}{0.5^2} = 192lx$$

ということになります。

最後に、ある面積をある方向から見たときに、その方向から放射される単位面積当たりの光度を**輝度**といいます。単位は*カンデラ毎平方メートル (cd/m^2)* または *ルーメン毎ストラジアン・平方メートル ($lm/sr \cdot m^2$)* です。

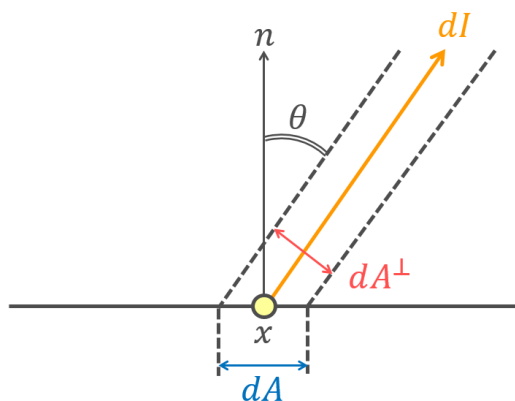


Fig.12: 輝度

点 x 周辺の微小面積 dA から、光度 dI 方向に出射する輝度 L を考えてみます。 dI 方向に垂直な微小面積 dA^\perp とすると、

$$dA^\perp = dA \cos \theta$$

という関係にあります。これは dI 方向からの微小面積 dA の見かけの面積です。この見かけの面積で除算すると単位面積当たりの光度を求めることができます。

$$L = \frac{dI}{dA^\perp} = \frac{dI}{dA \cos \theta}$$

光度 dI は単位立体角当たりの光束ですから

$$dI = \frac{d\Phi}{d\omega}$$

よって、輝度 L は

$$L = \frac{dI}{dA \cos \theta} = \frac{d\Phi}{dA \cos \theta \cdot d\omega}$$

となります。また、どの方向から見ても輝度が等しくなる面を**均等拡散面**といいます。このとき、法線方向の光度 dI_n と θ 方向の光度 dI_θ との間には

$$dI_\theta = dI_n \cos \theta$$

の関係があります。また、その光度の軌跡は円形となります。

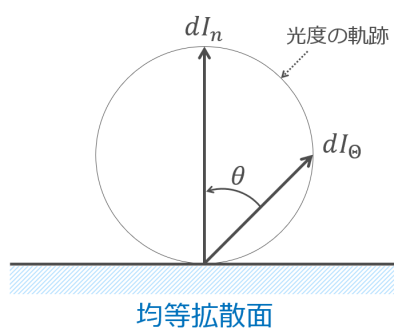


Fig.13: 均等拡散面

これを**ランベルトの余弦法則**といいます。また、照度とは対の関係で、ある単位面積当たりから出射する光束を**光束発散度**といい、 M と表記しますが、均等拡散面において、輝度 L と拡散発散度 M には次の関係があります。

$$M = \pi L$$

これはある面から半球に対して均等に放出されたものを積分 ($\cos \theta$ の半球積分) すると π となるからです。

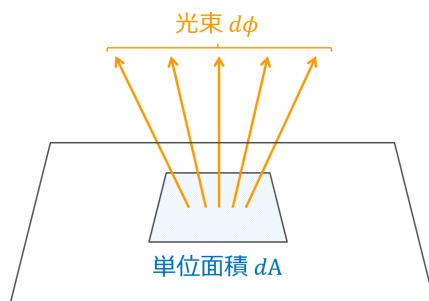


Fig.14: 光束発散度

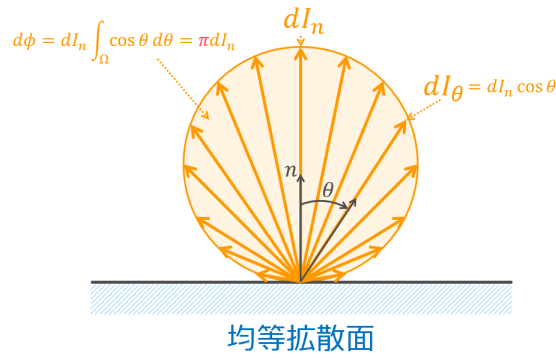


Fig.15: 均等拡散面の半球積分

均等拡散面が反射面とすると

$$M = \rho E$$

という関係があります。 ρ は反射率で、 E は照度です。 $\rho = 1.0$ の均等拡散面を完全拡散面といい、例えば石コウや白いチョークの表面がそれに近いと言われています。完全拡散面の場合、照度 E 、光束発散度 M 、輝度 L は

$$E = \pi \cdot L, \quad M = \pi \cdot L$$

の関係になります。

ここで、光度と照度のところで出てきた 60W の白熱電球を思い出してみましょう。このときの照度 E は 192lx でした。この面が完全拡散面なら光束発散度 M も $192lm/m^2$ となります。例えば、この面が白地で反射率 $\rho = 0.9$ の均等拡散面のとき、光束発散度 M は

$$M = \rho E = 0.9 \times 192 = 172.8lm/m^2$$

となります。そして、この面を見たときの輝度 L は

$$L = \frac{M}{\pi} = \frac{172.8}{\pi} = 55cd/m^2$$

となります。

これまで光束、光度、照度、輝度が出てきました。下図はこれらをまとめたものです。

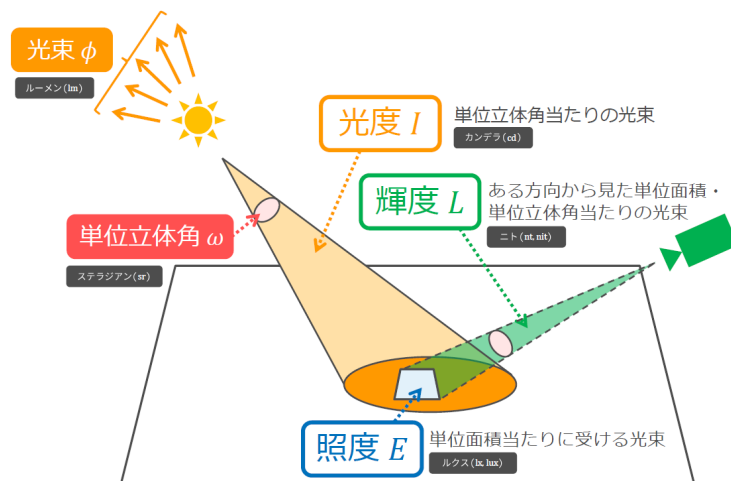


Fig.16: 光束・光度・照度・輝度の関係

また、それぞれの表記と単位を表にしました。

測光量	名称	表記	単位
光束	Luminous power	Φ	ルーメン (lm)
光度	Luminous intensity	I	カンデラ (cd) または lm/sr
照度	Illuminance	E	ルクス (lx, lux) または lm/m^2
輝度	Luminance	L	ニト (nt, nit) または cd/m^2

現実の光源による明るさの測光量は以下の資料がよくまとまっています。

明るさの参考		ライティング	
光源の種類	光束 [lm]	目安	照度 [lx]
ロウソク	20	満月の明かり	0.5
LED 3W	200	日没直後	5
ハロゲンランプ 40W	560	リビング(暗め)	70
ハロゲンランプ 100W	1400	家庭のトイレ	80
蛍光灯 40W	3000	ホテルのロビー	100
メタルハライド 200W	16000	リビング(明るめ)	500
発光体の目安	輝度 [cd/m^2]	オフィス	1000
日中の太陽	1.6×10^9	コンビニ	2000
地平線の太陽	600000	曇り	3000
白熱等	500000	晴天の日陰	10000
明るい空	10000	晴天の日向	100000
蛍光灯	10000	これらの値は実際にはばらつきが大きいのであくまでも目安です	
曇り空	3000		
月	2500		
TVモニタ	200		

Fig.17: 鈴木雅幸「リニア空間と物理的に正しいライティング」株式会社キャビア

レンダリングしたときの最終的な値は**輝度**になります。

5. 照明

光源は被写体を照らす光束を発生するものです。現実的に光源となるものとして太陽や月、白熱電球や蛍光灯、LED など様々なものがあります。これらの光源が放射した光束に被写体が直接照らされることを**直接照明 (Direct Illumination)**といいます。また、光源が放射した光束が他の物体に反射されて、被写体を照らすことを**間接照明 (Global Illumination, GI)**といいます。コンピュータグラフィックスでは、現実の光源を大胆に単純化したものが使われてきました。それは平行光源、点光源、スポットライトで、現在でもよく使われています。そして現実に近い光源として面光源、フォトメトリックライトがあります。間接照明についてはレイトレーシングやラジオシティ法、フォトンマップ法といったものや、リアルタイムで処理するようなもの、画像に光量を格納したものを使用するイメージベースドライティングなど様々な手法が開発されています。今回は直接照明として、平行光源、点光源、スポットライトを、間接照明としてイメージベースドライティングについて説明します。面光源とフォトメトリックライトについては軽く触れる程度とします。

ここでの目標はこれらの光源に対して、適切な単位を設定することです。残念ながらすべての光源の単位を統一することは逆にややこしくなってしまうため、それぞれ適切な単位を選択します。

5.1 照明の基礎

前の方で、光量について解説しました。ここではより詳しく光について見ていきます。

光は電磁波の一種です。人の眼に入ってきた電磁波は視神経を刺激し、それを明るさや色として感知します。つまり、光を色として認識しているのは人であって、光には色がありません。あくまで、人が色として認識するための性質を光が持っているということです。太陽光などの白色の光には様々な波長の電磁波が含まれています。光をプリズムに入射すると、7色の光に分光することができます。また、分光された光を逆向きにプリズムに入射させると元の白色光が得られます。プリズムでそれ以上に分光することができない光を**単色光**といい、波長順に並べたものを**スペクトル**といいます。人が認識できる光の波長は約 380～780nm ぐらいの範囲でこれを**可視光**といいます。

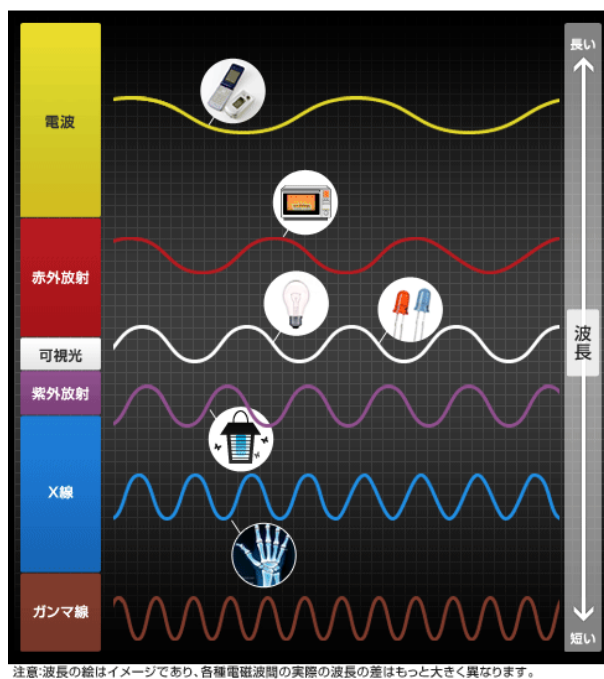


Fig.18: 波長. シーシーエス株式会社「光と色の話 第一部」より引用

https://www.ccs-inc.co.jp/guide/column/light_color/vol01.html

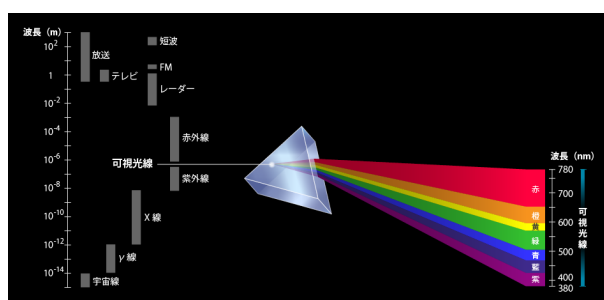


Fig.19: プリズム. コニカミノルタ「色の見え方と表現方法」より引用

<https://www.konicaminolta.jp/instruments/knowledge/color/section1/02.html>

光を見たときに明るさの感じやすさを**感度 (sensitivity)**といいます。また、感度を光の波長ごとに求めたものを**分光感度 (spectral sensitivity)**といいます。明るさの感覚は波長によって異なり、波長 555nm の黄緑色の部分は明るく感じ、380nm や 780nm 付近の青や赤の部分は暗く感じます。波長 555nm の明るさを 1 とし、これと同じエネルギーを持つ他の波長の明るさ感を比較値で表したものを**分光視感効率**といいます。分光視感効率には個人差があるので、多くの人の平均をとって国際照明委員会 (CIE) が**標準分光視感効率**を定めました。

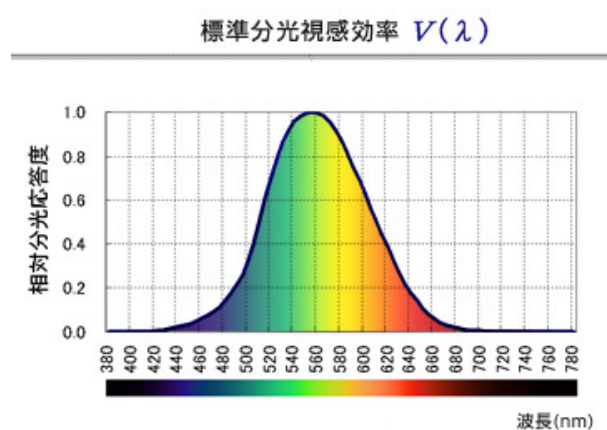


Fig.20: 標準分光視感効率. シーシーエス株式会社「光と色の話 第一部」より引用

https://www.ccs-inc.co.jp/guide/column/light_color/vol01.html

5.1.1 放射と光

電磁波あるいは粒子の形によって伝搬するエネルギーのことを**放射**といいます。この放射エネルギーの単位はジュール (J) で、 Q_e と表記します。放射量の場合は測光量と区別しやすいように記号に e をつけます。光束と同じように、単位時間当たりにある領域を通過する放射エネルギーを**放射束**といいます。放射束の単位はワット (W) で Φ_e と表します。

$$\Phi_e = \frac{dQ_e}{dt}$$

そして、放射束を目の感度（分光視感効率）に通した量が**光束**になります。ある放射体からの分光放射束が $\Phi_e(\lambda)$ 、標準分光視感効率が $V(\lambda)$ で与えられれば、光束 Φ は

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

となります。ここで λ は波長、 K_m は最大視感度で、その値は約 555nm において約 683lm/W となります。

5.1.2 順応

人の目というのは高性能で、例えば月明りの 0.1lx から直射日光の 10 万 lx に至るまでのダイナミックレンジに対応しています。目が明るいところや暗いところでも見れるように調整してくれます。例えば一瞬まぶしく感じたところでも時間が経つと徐々に自然な感じで見れるようになります。これを**順応**といいます。明るいところに順応することを**明順応**、暗いところに順応することを**暗順応**といいます。一般に、目が完全に暗順応するのに 30 分以上かかるのに対し、明順応は 1 分以内で完了すると言われています。

5.1.3 演色性

物体の色の見え方が光源によって変わることがあります。色の見え方には、その物体を照明する光の特性（分光分布）、物体自身の光学的特性（分光反射率）、人の視覚特性の 3 つの要素が関係しています。色の見え方が変わる要因として、光源の分光分布が変化したことと、人の目が照明光や周囲の色度に目が慣れてしまい、目の感度が変わることです。前者で本来の色とずれた分、後方で調整しようとしませんが、完全に調整できず、結果として色の見え方が変わってしまいます。照明光が物体の色の見え方に及ぼす影響を**演色**といい、この演色効果を決める光源固有の性質を**演色性**といいます。光源の演色性は**平均演色評価数** (R_a) および**特殊演色評価数** ($R_9 \sim R_{15}$) によって表されます。この平均演色評価数が 100 に近いほど演色性が良くなります。

5.1.4 発光

光を発生させるための方法として**熱放射**と**ルミネセンス**によるものの2つがあります。熱放射は物体を熱したときに、原子、分子、イオンなど物質粒子の熱振動によってエネルギーが放出される現象です。低温だと赤外線が放出されますが、高温になると可視光が放出されるようになります。白熱電球はフィラメントを使用して電氣的に高温にすることで発光しています。次に、ルミネセンスは物体が光、放射、電子、電界などのエネルギーを吸収し、それが放射エネルギーを放射する現象です。この現象を発生させるために、物体に何らかの刺激を与える必要があります、その方法には様々なものがあります。

5.1.5 照明光源

照明用の光源には HID ランプ、ハロゲン電球、白熱電球、蛍光灯、LED といった様々なものがあります。ハロゲン電球と白熱電球は熱放射による発光、HID ランプや LED は電界によって励起させるエレクトロルミネセンス、蛍光灯は光によって励起するホトルミネセンスによって発光しています。

市販されている照明には消費電力 (W) や光束 (lm) が記載されています。消費電力から光束を求める場合は、その発光効率 η (単位電力当たりの全光束 lm/W) から

$$\Phi = \eta \cdot \Phi_e$$

で求めることができます。また、視感度 V を使用することもできます。

$$\Phi = 683 \times \Phi_e \times V$$

例としていくつかの照明とその発光効率、視感度を表にしました。

照明のタイプ	発光効率	視感度
白熱電球	14-35	2-5%
LED	28-100	4-15%
蛍光灯	60-100	9-15%

詳細は Wikipedia の [発光効率](#) を参考にしてください。

5.2 直接照明

直接照明の場合は、光源から対象の領域の照度を求めて、輝度を計算します。対象の領域からある方向に出射される輝度 L は、照度 E 、BSDF $f(v, l)$ として

$$L_{out} = f(v, l) E$$

で求められます。照度は半球上のあらゆるところから入ってくる輝度を積分したものですので、出射方向の立体角 ω' 、入射方向の立体角 ω とすると

$$L_{out}(\omega') = \int_{\Omega} f(\omega', \omega) dE(\omega') = \int_{\Omega} f(\omega', \omega) L_{in}(\omega) (\omega' \cdot n) d\omega$$

となります。

5.2.1 平行光源

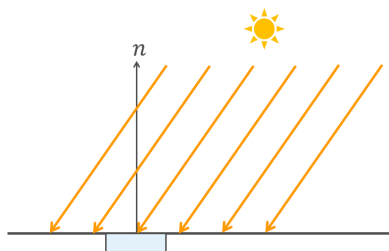


Fig.21: 平行光源

平行光源は方向のみで表される光源です。この光源は現実にはありませんが、計算が一番軽いということもあって太陽や月の光を表現するのによく用いられます。平行光源には**照度** (lx) を指定します。それは、公開されている空や太陽、月といった光束の値が照度になっているということと、実際の計算でそのまま使えるからです。平行光源による輝度は

$$L_{out} = f(v, l) E_{\perp} \langle n \cdot l \rangle$$

ここで E_{\perp} は、その面に垂直に入射した照度で、 $\langle n \cdot l \rangle$ はコサイン項と呼ばれ、入射角余弦の法則によるものです。実際に計算するコード (GLSL) は次のようになります。

```
vec3 l = normalize(-lightDirection);  
float NoL = clamp(dot(n,l), 0.0, 1.0);  
float illuminance = lightIntensity * NoL; // lightIntensity = lx  
vec3 luminance = BSDF(v,l) * illuminance;
```

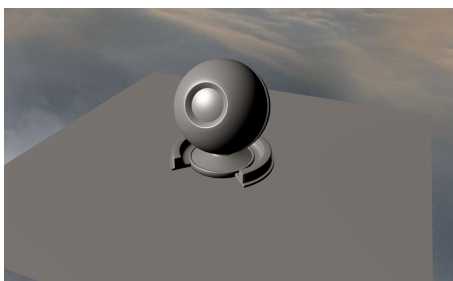


Fig.22: 平行光源

5.2.2 点光源

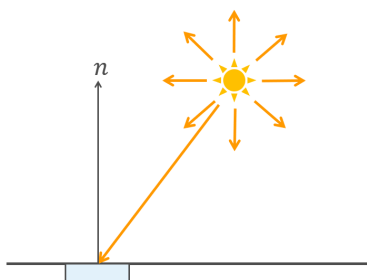


Fig.23: 点光源

点光源はある位置から全方向に対して同じ光度を放射する光源です。これも現実には存在しませんが、光源がとても小さい場合には近い照明効果が得られます。こちらと比較的負荷が軽く、調整がしやすいため、リアルタイムレンダリングなどでは一番使われると思います。点光源についてはこれまでに何度か出ているのでわかりやすいと思います。まず、点光源から放出される全光束は

$$\Phi = \int_{\Omega} I d\omega$$

です。実際に計算するときは立体角を極座標に変換します。すると

$$\Phi = \int_{\Omega} I dl = \int_0^{2\pi} \int_0^{\pi} I d\theta d\phi = 4\pi I \quad \therefore I = \frac{\Phi}{4\pi}$$

となります。照度 E は

$$E = L_{in} \langle n \cdot l \rangle = \frac{I}{d^2} \langle n \cdot l \rangle$$

の関係なので輝度は

$$L_{out} = f(v, l) E = f(v, l) \frac{I}{d^2} \langle n \cdot l \rangle = f(v, l) \frac{\Phi}{4\pi d^2} \langle n \cdot l \rangle$$

と計算できます。点光源では**光束 (lm)** を指定します。

点光源は逆 2 乗の法則によって減衰しますが、実際にこれが扱いづらいことがあります。まず、プログラムのなことでゼロ除算が発生すること場合があります。次に、距離が極端に短いと光束がかなり大きくなってしまいます。また、逆 2 乗はゼロに収束しません。

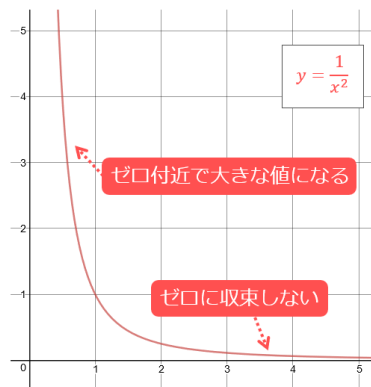


Fig.24: 逆 2 乗

まず収束しない問題については影響する半径を設定し逆 2 乗を窓にします。

$$\frac{1}{d^2} \left\langle 1 - \left(\frac{d}{r} \right)^4 \right\rangle^2$$

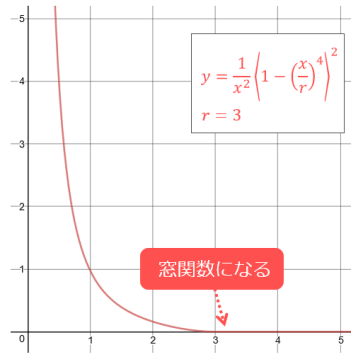


Fig.25: 逆 2 乗の窓関数化

次にゼロ付近を緩やかにします。

$$\frac{1}{d^2 + 1} \left\langle 1 - \left(\frac{d}{r}\right)^4 \right\rangle^2$$

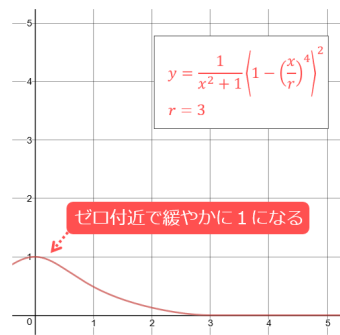


Fig.26: 逆 2 乗の緩やかに 1 にする

また、+1 したことでゼロ除算しなくなります。 r についてはゼロ除算する可能性があるので、とても小さい値を設定します。この減衰を使って実際に計算するコード (GLSL) は次のようになります。

```
float getSquareFalloffAttenuation(float lightDistance, float lightRadius) {
    float sqDistance = lightDistance * lightDistance;
    float invRadius = 1.0 / lightRadius;
    float factor = sqDistance * invRadius * invRadius;
    float smoothFactor = max(1.0 - factor*factor, 0.0);
    return (smoothFactor * smoothFactor) / (sqDistance + 1.0);
}

...

float NoL = clamp(dot(n,l), 0.0, 1.0);
float illuminance = lightIntensity * NoL; // lightIntensity = lm / 4pi
float attenuation = getSquareFalloffAttenuation(lightDistance, lightRadius);
vec3 luminance = BSDF(v,l) * illuminance * attenuation;
```

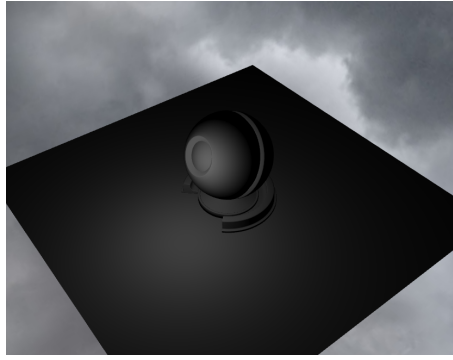


Fig.27: 点光源

5.2.3 スポットライト

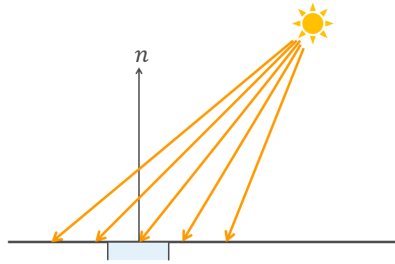


Fig.28: スポットライト

スポット光源は指向性をもった点光源のようなもので、円錐状の照明をします。これも現実には存在しませんが、サーチライトや車のフロントライトなどをシミュレーションするために使われたりします。スポット光源は $[0..\pi]$ の範囲を持つ円錐の外角 θ_{outer} と内角 θ_{inner} を指定します。スポット光源から放射される全光束 Φ は

$$\Phi = \int_{\Omega} I d\Omega = \int_0^{2\pi} \int_0^{\theta_{outer}} I \sin\theta d\theta d\phi = 2\pi \left(1 - \cos\frac{\theta_{outer}}{2}\right) I$$

$$\therefore I = \frac{\Phi}{2\pi \left(1 - \cos\frac{\theta_{outer}}{2}\right)}$$

となります。これから輝度 L_{out} は

$$L_{out} = f(v, l) \frac{\Phi}{2\pi \left(1 - \cos\frac{\theta_{outer}}{2}\right) d^2} \langle n \cdot l \rangle \lambda(l)$$

になります。こちらも減衰については逆2乗ではなく点光源のところで導出したものを使用します。 $\lambda(l)$ は減衰項で

$$\lambda(l) = \frac{l \times spotDirection - \cos\theta_{outer}}{\cos\theta_{inner} - \cos\theta_{outer}}$$

スポット光源も点光源と同様に**光束 (lm)** を指定します。この計算コード (GLSL) は次のようになります。

```
float getSquareFalloffAttenuation(float lightDistance, float lightRadius) {
    float sqDistance = lightDistance * lightDistance;
    float invRadius = 1.0 / lightRadius;
    float factor = sqDistance * invRadius * invRadius;
    float smoothFactor = max(1.0 - factor*factor, 0.0);
    return (smoothFactor * smoothFactor) / (sqDistance + 1.0);
}
```

```

float getSpotAngleAttenuation(float angleCos, float innerAngleCos, float outerAngleCos) {
    float spotScale = 1.0 / max(innerAngleCos - outerAngleCos, 1e-4);
    float spotOffset = -outerAngleCos * spotScale;
    float attenuation = clamp(angleCos * spotScale + spotOffset, 0.0f, 1.0);
    return attenuation * attenuation;
}

...

float angleCos = dot(lightDirection, spotLightDirection);
float NoL = clamp(dot(n,l), 0.0, 1.0);
float illuminance = lightIntensity * NoL; // lightIntensity = lm / (2pi * (1-cos(outerAngle*0.5)))
float attenuation = getSquareFalloffAttenuation(lightDistance, lightRadius);
attenuation *= getSpotAngleAttenuation(angleCos, innerAngleCos, outerAngleCos);
vec3 luminance = BSDF(v,l) * illuminance * attenuation;

```



Fig.29: スポットライト

この光度の計算は物理的に正しいですが、照度がスポットライトの外角によって変わるので扱いづらいかもしれません。そこで、以下のようにスポットライトの外角に依存しないようにすることができます。

$$\Phi = \pi I \quad \therefore I = \frac{\Phi}{\pi}$$



Fig.30: スポットライト：外角に影響しない

5.2.4 面光源

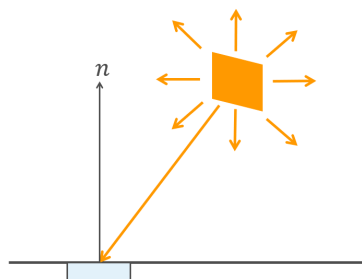


Fig.31: 面光源

現実の光源は大きさ（面積）を持っています。面光源は微小面積を点光源と考えて、光源面を積分して求めます。計算が他のと比べて多いのですが、現実と同じ照明に近く、リアルタイムレンダリングでも使うことが多くなってきています。これも点光源とスポットライトと同じ**光束 (lm)**を指定します。今回は詳しく扱いません。

5.2.5 フォトメトリックライト

演色性のところで出てきましたが、光源の分光分布は単純なものではなく、製品によって異なります。販売されている照明機器にはその分光分布が含まれているフォトメトリックファイルが公開されており、それを使用することで正確な光源の特性を再現することができます。フォトメトリックファイルの1つとしてIESファイルには光源周りの様々な角度における光度が入っています。

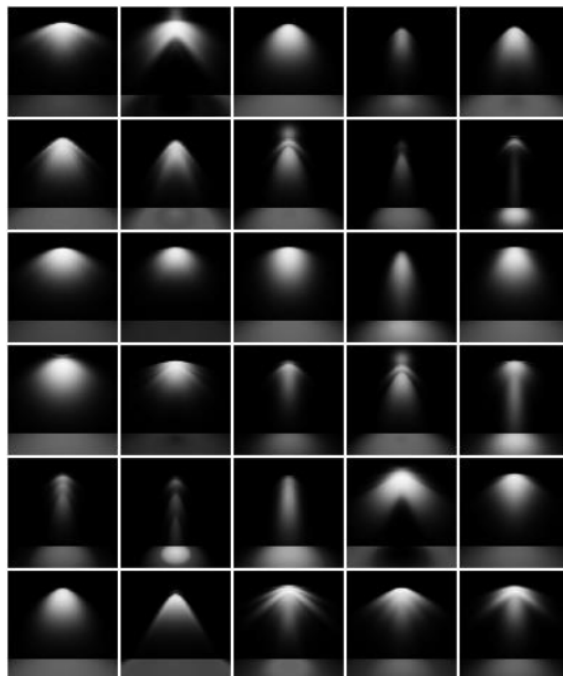


Fig.32: フォトメトリックライト。Filamentのドキュメントより引用 <https://google.github.io/filament/Filament.md.html>

フォトメトリックライトでは**光度** (cd) を使います。今回は詳しく扱いません。

5.3 間接照明

現実では光源から放射された光が物体に反射して、私たちの目やカメラに届く前に、別の物体に反射したり吸収されたりします。物体の周辺にある環境全体が光源であり、それは何度も物体に反射された光だったりします。直接照明ではなく、他の物体に反射された光が光源として照明を行うことを**間接照明**といいます。間接照明には様々な手法が開発されていて、リアルタイムレンダリングに特化したものや、光の反射をトレースするものがあります。

5.3.1 IBL

間接照明の1つとしてイメージベースドライティングと呼ばれるものがあります。これは物体周辺の反射された光をあらかじめ計算しておいて画像として保存しておき、レンダリングするときに使う方法です。リアルタイムやオフラインレンダリング両方でよく使われています。画像には全方向の**輝度** (nt) が格納されています。実際には拡散反射、鏡面反射用に別々に事前計算したものを利用します。拡散反射の場合には画像ではなく球面調和関数の係数にすることでデータ量を劇的に改善しています。実際の計算は輝度から照度を求めて直接照明と同じように BSDF によって出射する輝度を求めます。

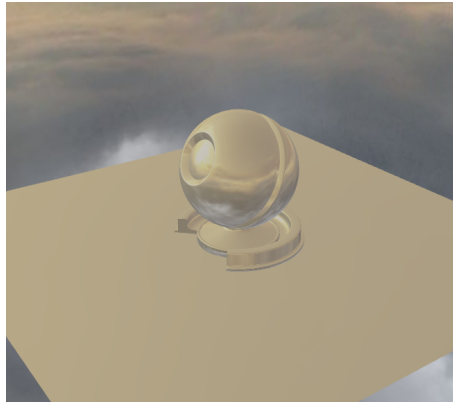


Fig.33: IBL

5.4 照明の単位

直接照明と間接照明について、それぞれ使用する単位と使い方を説明してきました。ここで表にまとめておきます。

光源の種類	光量	単位
平行光源	照度	lx
点光源	光束	lm
スポットライト	光束	lm
面光源	光束	lm
hline フォトメトリックライト	光度	cd
IBL	輝度	nt

5.5 計測

それぞれの光源の種類に対して適切な照明の単位を選びました。次に、その値をどうやって調べるかということですが、まず、市販されている照明については全光束量がほぼ公開されています。次に太陽光といった照度はインターネットなどで調べることができます。最後にそれでもわからない場合は計測することになります。照度や輝度の光量計が販売されていますので、そちらを購入することになります。お手頃な価格から、かなりお高いものまで様々な種類がありますが、Sekonic 製のものがよく使われているようです。光源の光度については照度と計測機器との距離から

$$E = \frac{I}{d^2} \quad \therefore I = \frac{E}{d^2}$$

で得られます。

6. 色

光そのものに色というものはなく、人が色と認識できる性質を光が持っているということは照明の基礎のところでも言及しました。光源の正確な色を再現する場合は光源の分光分布に基づいてスペクトルを取り出して RGB に変換することになります。現実問題としてそのような処理は負荷が高く、分光分布のデータも必要なので、光源の色を RGB で直接指定するのが現状だと思います。また、照明においては光源の色を表すものとして色温度があります。これについて見ていくことにしましょう。

6.1 色温度

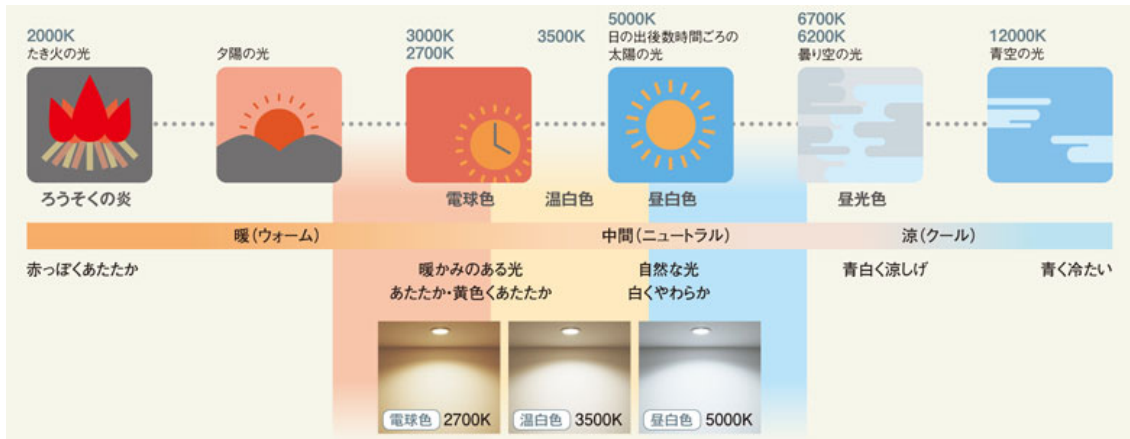


Fig.34: 色温度. (c) パナソニック・エコソリューションズ

色温度は簡潔に言うと、通常、色を表す値として RGB の3つが必要なところ、**絶対温度**という1つの値で色を表すことができるものです。もちろん、この絶対温度から RGB に変換することになります。この絶対温度ですが、温度ということから熱に関係していることは想像できると思います。照明の発光のところで熱放射というのが出てきたのを思い出してください。一般に、鉄などの不燃性物体を加熱すると、次第に赤く光ってきます。さらに加熱して高温になると赤色、黄色、白色へと変化していきます。ここで、最も理想的な物体で現実には存在しない**黒体**というのを考えます。この黒体は入射した光をすべて吸収する（反射しない）物体で**完全放射体**といいます。つまり黒体から放射される光は必ず黒体自身が発光したものと考えられます。この黒体の熱放射を特に**黒体放射**といいます。そして、ある放射体の光色に等しい光色を持つ黒体の温度を、その放射体の色温度といいます。実際には完全に等しい光色の色温度がない場合、最も近い温度が使われます。話を少し戻して、絶対温度というのはケルビン温度ともいい、数値のあとに K をつけて表します。例えば色温度 6000K のように使います。照明機器や現実の色光について色温度として表されていることがありますので、実際にこの値を使うためには RGB に変換しなければなりません。そのために、RGB や XYZ といった色空間について知る必要があります。

色を扱うために色の基本から説明したほうがよいのですが、用語のオンパレードで、正直うんざりするので、必要最低限のものだけ取り扱います。詳しく知りたい方は色彩工学の資料を参照してください。

6.2 色空間

色を指定するうえで基本となるのは RGB で、赤色・緑色・青色の3つです。この3つを適当な割合で混ぜ合わせると任意の色を作ることができます。この3つの色は**光の三原色**と呼ばれていますね。混ぜ合わせて色を作るので**加法混色**ともいいます。また、赤色、緑色、青色の割合を値で表しますが、各々を**刺激値**といい、3つ合わせて**三刺激値**と呼びます。そして人の目は入ってきた光が RGB の3種の視細胞（錐体）を刺激して知覚します。この光を**色刺激**といいます。この色刺激と RGB の三刺激値（色刺激に対して原刺激といいます）が一致するように混合の割合を調整することを**等色**といいます。

一定の放射エネルギーで波長 λ の単色光の色刺激を**単色光刺激**または**スペクトル刺激**といい、単色光刺激 F_λ を原刺激 R, G, B の混合で等色するとき、その混合量を $\bar{r}_\lambda, \bar{g}_\lambda, \bar{b}_\lambda$ とすれば

$$F_\lambda = \bar{r}_\lambda R + \bar{g}_\lambda G + \bar{b}_\lambda B$$

となります。このときの混合量を**等色係数**といい、波長の関数として求めた等色係数 $\bar{r}(\lambda), \bar{g}(\lambda), \bar{b}(\lambda)$ を**等色関数**といいます。ま

た、この式を**等色式**といいます。等色式の原因刺激を単位ベクトルとすれば、3次元空間の位置が決まります。この色の3次元空間を**色空間**といいます。以降は基本的に CIE が定めた色空間を指しています。

等色関数がわかれば、光の波長から RGB が求まります。下図は RGB 色空間の等色関数のグラフです。波長によって RGB それぞれの値が求まりますので、光に含まれる波長を等色関数で RGB を求めて積分することで、その光の色を RGB に変換することができます。

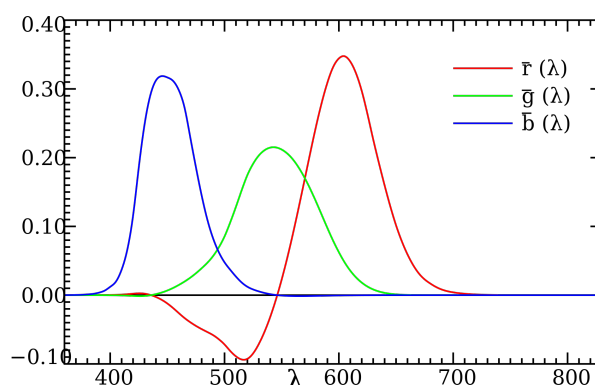


Fig.35: CIE RGB 等色関数. Wikipedia 「CIE 1931 色空間」より引用

グラフを見てみるとある波長で赤が負の値になっていることがわかります。負の値があると扱いづらいということで、等色関数がすべて正の値になるように新しい原刺激 X, Y, Z を CIE が決めました。それが XYZ 色空間です。

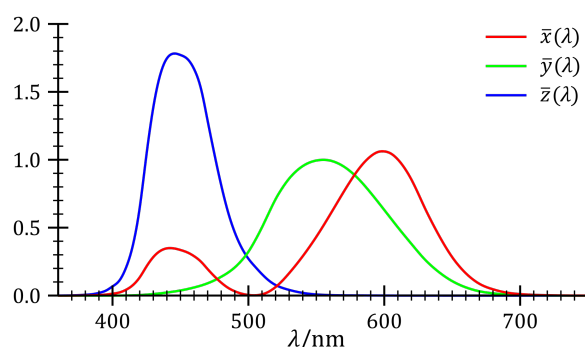


Fig.36: CIE XYZ 等色関数. Wikipedia 「CIE 1931 色空間」より引用

色空間は3次元空間ですが、3次元よりも2次元のほうがわかりやすいということで、色空間を2次元に投影します。単位ベクトルですので $X + Y + Z = 1$ が成り立ちます。これは3次元空間上の平面上に存在しているので、それを XY 平面に投影します。

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

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

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

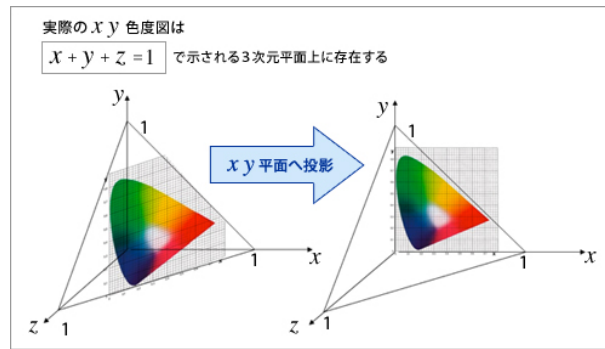


Fig.37: xy 空間に投影. シーシーエス株式会社「光と色の話 第一部」より引用

https://www.ccs-inc.co.jp/guide/column/light_color/vol29.html

このように2次元平面に投影した色空間の図を**色度図**といい, x, y を**色度座標**といいます. 下図は xy 色度図です.

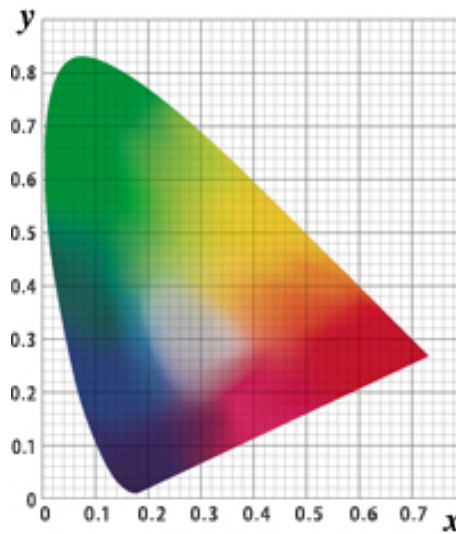


Fig.38: xy 色度図. シーシーエス株式会社「光と色の話 第一部」より引用

https://www.ccs-inc.co.jp/guide/column/light_color/vol29.html

XYZ 空間の Y は輝度を表します. そこで, 色度座標 x, y と Y を用いた **xyY 色空間**というのがあります. X, Z の刺激値は

$$X = \frac{Y}{y} x$$

$$Z = \frac{Y}{y} (1 - x - y)$$

で求めることができます.

単色光の色度座標を繋げた曲線を**単色光軌跡**または**スペクトル軌跡**といいます.

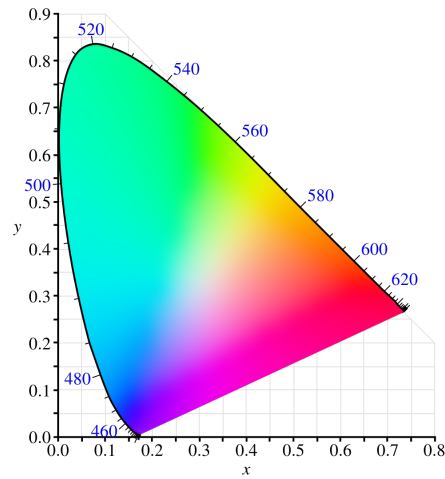


Fig.39: 単色光軌跡. シーシーエス株式会社「光と色の話 第一部」より引用

https://www.ccs-inc.co.jp/guide/column/light_color/vol29.html

6.3 XYZ と RGB の相互変換

行列とベクトルを使って表すと、RGB(sRGB) 空間から XYZ 空間への変換は次のようになります。

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0.412391 & 0.357585 & 0.180482 \\ 0.212639 & 0.71517 & 0.0721926 \\ 0.0193308 & 0.119195 & 0.0950536 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

そして、XYZ 色空間から RGB(sRGB) 色空間への変換は次のようになります。

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 3.24097 & -1.53738 & -0.498612 \\ -0.969241 & 1.87596 & 0.0415554 \\ 0.0556299 & -0.203976 & 1.05697 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

6.4 色温度から RGB 色空間へ

まず、単色光軌跡のように、黒体温度の xy 座標を繋げた曲線を**黒体軌跡**といいます。

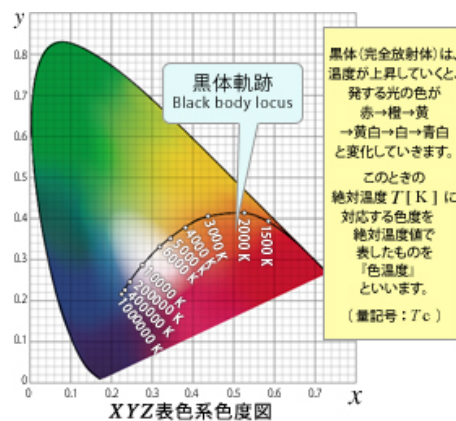


Fig.40: 黒体軌跡. シーシーエス株式会社「光と色の話 第一部」より引用

https://www.ccs-inc.co.jp/guide/column/light_color/vol33.html

昼光（太陽光）は最も重要な自然光源で、その分光分布について多くの測定が行われた結果、測定するときの場所・時刻・天候などの要因で変動することがわかりました。そこで CIE はその結果を解析して、xy 色度図上に昼光の色度座標 x_D, y_D を定めました。

$$y_D = -3.000x_D^2 + 2.870x_D - 0.275$$

x_D の値は**相関色温度** T_{cp} (黒体温度に一致しないときの最も近い色温度) の範囲によって変わります。 $4000K \leq T_{cp} \leq 7000K$ のとき

$$x_D = -4.6070 \cdot 10^9 / T_{cp}^3 + 2.9678 \cdot 10^6 / T_{cp}^2 + 0.09911 \cdot 10^3 / T_{cp} + 0.244063$$

であり、また、 $7000K \leq T_{cp} \leq 25000K$ では

$$x_D = -2.0064 \cdot 10^9 / T_{cp}^3 + 1.9081 \cdot 10^6 / T_{cp}^2 + 0.24748 \cdot 10^3 / T_{cp} + 0.237040$$

となります。この得られた平均的な昼光を **CIE 昼光** といいます。また、xy 色度図上の軌跡を**昼光軌跡**といい、黒体軌跡に近い値となっています。

6.5 標準の光源

物体の色は三刺激値 X, Y, Z で規定されますが、照明光の分光分布にも依存します。そのため、CIE は標準的な照明光を定めました。その中で昼光を代表するものとして相関色温度 6500K の昼光 D_{65} 、また、相関色温度がそれぞれ 5000K, 5500K, 7500K の昼光 D_{50}, D_{55}, D_{75} があります。この値も市販されている製品などで使われています。

7. カメラ

まずは次の画像を見てください。この画像覚えているでしょうか。



Fig.41: 光源・被写体 (マテリアル)・CG カメラ

最初の方に出てきたものですが、この中で光源のところをこれまで説明してきました。被写体 (マテリアル) の部分は簡単にいうと物体の反射に関する内容で、「基礎からはじめる物理ベースレンダリング」や「拡散 BRDF」「鏡面 BRDF」で言及しているもので、今回はこの部分は扱いません。ですので、残りは CG カメラということになります。

光源のところで説明したように現実の太陽光の照度や市販されている光束の値を使って計算を行うと、明らかに値が大きくなっています。例えば日中の太陽光を他のと一緒に扱うと 32bit の浮動小数では扱いきれません。なので、露出補正を行ってダイナミックレンジを調整することが一番の目的です。

露出補正の話に進む前に、少しカメラについて説明します。まずカメラは光を受けとって撮像素子に作用させて電気信号に変換して、画像データとしてメモリーに保存します。カメラは複数のレンズを通して入射した光を 1 箇所に集めます。その位置を**焦点**といい、ここに像をつくります (結像)。また、この焦点から光学的な中心 (主点) までの距離を**焦点距離**といいます。この焦点距離が 2 倍のレンズを使えば 2 倍の大きさの像を作ります。焦点の明るさはレンズの口径に依存していて、口径が大きければ入射する光も多くなり明るくなります。

7.1 絞り、シャッタースピード、ISO 感度

露出補正に関係のあるパラメータは絞り値、シャッタースピード、ISO 感度です。それぞれ簡単に説明していきます。まず、絞り値ですが、これは絞りの値です。絞りは、複数枚の羽根で同心円状などの形状を保ちながら、レンズに入射する光量を調整するもので、絞り径を表す **F 値** で指定します。F 値は、F1 を基準に $\sqrt{2}$ 倍するごとに、絞り径が 2 倍、像面照度が半分（つまり光量が半分）になっていきます。例えば F1, F1.4, F2, F2.8, F4, F5.6, F8, F11, F16 となっています。カメラで焦点を調整するときはこの絞り、焦点距離、撮影距離が関係してきます。詳しくは解説しませんが、使用しているレンズの最小の F 値（これを開放絞り値といいます）まで開くと、焦点のあった部分だけが鮮明になり、それ以外がボケてしまいます。つまり被写界深度が浅いです。逆に最大の F 値まで絞ると鮮明に被写界深度が深くなります。そして絞るほど光量が減少していきます。

次にシャッタースピードです。これはシャッターが開く時間で、撮像素子に光を作用させる時間です。この時間が短いと光量が少ないので暗くなります。逆に長いと、被写体が速く動いていれば被写体ブレ、特に手で持って撮影したときに手ブレが発生しやすくなります。また、光量が多くなりますので、明るくなります。手ブレについては三脚で固定するといった対応ができます。被写体ブレは逆に光を蓄積して被写体の状態や軌跡を表現することができます。シャッタースピードは 1 秒を基準として、2, 4, 8, 15, 30, 60, 125, 250, 500, ... といった基本的に 2 の倍数となって、2 なら 1/2 秒ということになります。

最後に ISO 感度です。これは撮像素子が電気信号に変換する時に光を増幅します。100 を基準として 2 倍した値になるごとに増幅率も 2 倍されます。ISO 感度を上げすぎるとノイズが発生するという問題があります。絞りやシャッタースピードを調整して、それでも明るさが足りない時に ISO 感度を上げるのが基本です。

7.2 露出値

撮像素子に作用する光量を表す値として **露出値 (Exposure Value: EV)** があります。これは絞り値とシャッタースピードで決まります。絞り値を **AV (Aperture Value)**、シャッタースピードを **TV (Time Value)** とすると、露出値 EV は

$$EV = AV + TV$$

という関係にあります。ここで、AV は F1.0 のとき AV0 で、F2 のとき AV1、F2.8 のとき AV3 です。また、TV は 1 秒が TV0 で、1/2 秒が TV1、1/4 秒が TV2 となっています。そのため、F 値を N 、シャッタースピードを t としたとき、次のような計算で EV を計算できます。

$$EV = \log_2 \left(\frac{N^2}{t} \right)$$

露出値を +1 すれば、2 倍の光量となり、-1 すれば半分の光量となります。また、シャッタースピードを 2 倍にすれば EV は +1 で、F 値を 2 倍にすると EV は +2 になります。

7.3 露出補正

それでは、3つの値を使ってダイナミックレンジを調整する露出補正をします。求めたいのは光源のところで実際のコードをいくつか提示しましたが、その中の **lightIntensity** に対して乗算します。つまり

```
lightIntensity *= exposure;
```

とします。実際には CPU 側で露出補正值 **exposure** を算出し、シェーダに送る前に事前に乗算しておくことで負荷を軽減します。

まずは、露出値 EV の式から考えます。

$$EV = \log_2 \left(\frac{N^2}{t} \right)$$

露出値 EV は 1 増えると光量も 2 倍になりましたね。そして、ISO 感度も 2 倍になると増幅されて 2 倍になります。この関係を式にすると、ISO100 の露出値を EV_{100} として、任意の感度 S の露出値 EV_S は

$$EV_S = EV_{100} + \log_2 \left(\frac{S}{100} \right)$$

となります。 N, t, S を使って EV_{100} を求める式に書き直すと

$$EV_{100} = EV_S - \log_2 \left(\frac{S}{100} \right) = \log_2 \left(\frac{N^2}{t} \right) - \log_2 \left(\frac{S}{100} \right)$$

となります。これは任意の感度 S を影響させた上で、ISO 感度 100 の露出値としています。次にこのカメラの測光輝度の飽和値を算出して、その値を使って正規化し、ダイナミックレンジを調整します。まず、カメラの像面露光量 $H(lx \times s)$ は

$$H = \frac{q \cdot t}{N} L$$

です。 q は

$$q = \frac{\pi}{4} T \nu(\theta) \cos^4 \theta = 0.65$$

となっていて、 T はレンズの透過率に依存する補正係数、 $\nu(\theta)$ はレンズの減光（ビネッティング）係数、 θ はレンズの軸に対する角度で、一般的に $\theta = 10^\circ$ 、 $T = 0.9$ 、 $\nu = 0.98$ から求めると 0.65 という値になります。像面露光量を求める式には感度 S が含まれていないため、Saturation-based speed の方式を使います。

$$H_{sat} = \frac{78}{S_{sat}}$$

H_{sat} は最大露光量です。この式を像面露光量の式に統合して、最大輝度 L_{max} を求めると

$$L_{max} = \frac{N^2}{q \cdot t} \frac{78}{S}$$

最大輝度を使えば入社輝度 L を正規化することができます。

$$L' = \frac{L}{L_{max}}$$

L_{max} は、 $S = 100, q = 0.65$ を代入すると

$$\begin{aligned} L_{max} &= \frac{N^2}{q \cdot t} \frac{78}{S} \\ &= 2^{EV_{100}} \frac{78}{q \cdot S} \\ &= 2^{EV_{100}} \times 1.2 \end{aligned}$$

となって簡単になります。これをコードにすると

```
float getEV100(float aperture, float shutterSpeed, float sensitivity) {
    return log2((aperture*aperture) / shutterSpeed * 100.0f / sensitivity);
}

float getExposure(float ev100) {
```

```

    return 1.0f / (pow(2.0f, ev100)*1.2f);
}

float ev100 = getEV100(aperture, shutterSpeed, sensitivity);
float exposure = getExposure(ev100);

```

となって、露出補正値が求まります。

7.4 事前露出補正

本来はカメラに入ってくる時（フラグメントシェーダの出力時）に露出補正を行うものですが、あらかじめ光源の光量にこの露出補正をしておきます。これは余計なコストを削除する他に、浮動小数の精度の問題で、先に補正しておかないと大きい値となってまともに取り扱えません。光源の光量を露出補正して、かつ先に計算できるところはやってしまいましょう。まず平行光源ですが、これは露出補正のみです。

```
dirLight.intensity = lx * exposure;
```

次に点光源です。これは 4π を除算しておきます。

```
pointLight.intensity = lm * (1/(4*PI)) * exposure;
```

スポットライトは2種類あります。

```

float denom = 1.0f / (2.0f * PI * (1.0f - cos(outerAngle) * 0.5f));
spotLight.intensity = lm * denom * exposure;

// and

spotLight.intensity = lm * (1/PI) * exposure;

```

あとは、IBL なのですが、これは実装方法によって対応が変わるので、事前に計算できる場合は先に計算しておきます。

8. さいごに

あまり検証出来ずに書いてしまいました。間違いなど見つけたらご連絡いただけると嬉しいです。参考になれば幸いです。

参考文献

- [1] 「Filament」 <https://github.com/google/filament>
- [2] Brian Karis 「Real Shading in Unreal Engine 4」, 2013
- [3] 鈴木雅幸 「リニア空間と物理的に正しいライティング」, 2008
- [4] 橋口智仁, 村岡伸一 「カメラ的に正しいフォトリアルグラフィック制作フロー」, 2014
- [5] 内村創 「ゲームのための色彩工学」, 2016
- [6] 齊藤修 「大規模 CS ゲームにおけるライトマス運用」, 2016
- [7] 「CIPA DC-004-2004 デジタルカメラの感度規定」, 2004
- [8] CG-ARTS 協会 「コンピュータグラフィックス」 CG-ARTS 協会, 2015
- [9] 堀内敏行 「光技術入門」 東京電機大学出版局, 2014
- [10] 大田登 「色彩工学」 東京電機大学出版局, 2014
- [11] 一般社団法人 照明学会 「照明工学」 オーム社, 2012
- [12] 豊田堅二 「カメラメカニズム講座」 日本カメラ社, 2017
- [13] 大和田良, 勝倉峻太, 岸剛史, 木村崇志, 船生望, 圓井義典 「写真の教科書」 インプレス, 2016
- [14] 中原一雄 「写真のことが全部わかる本」 インプレス, 2018