

7. ハードライト(Hardlight)

計算式は以下のようになります。一見オーバーレイと同じように見えますが、重ねられるレイヤーではなく、重ねるレイヤー(上のレイヤー)の中間値で乗算とスクリーンを使い分けています。逆に言えば、重ねるレイヤーと重ねられるレイヤーを入れ替えると、オーバーレイと全く同じ結果が得られます。

Pb < 128の時

$$P_n = P_a * P_b * 2 / 255$$

Pb >= 128の時

$$P_n = 2 * (P_a + P_b - P_a * P_b / 255) - 255$$

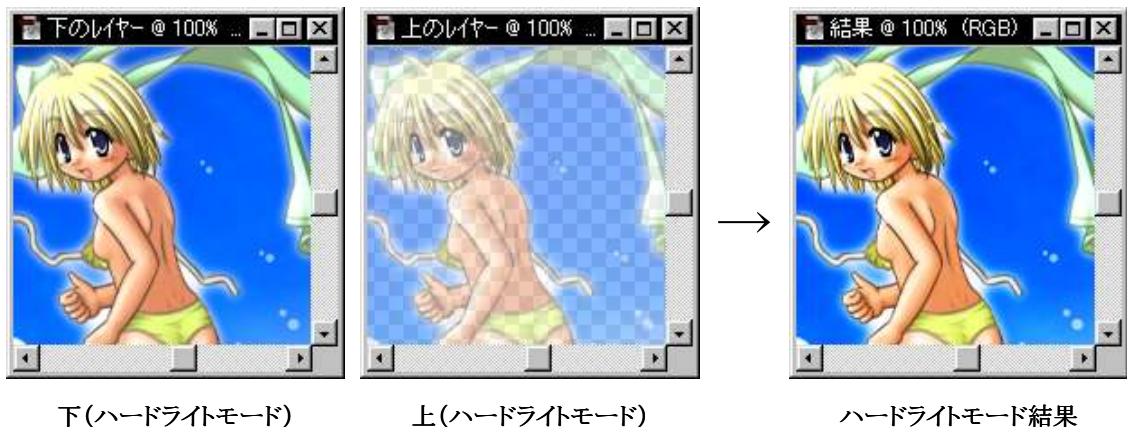
Pn、Pa、Pb を0～1の範囲とする場合、

Pb < 0.5の時

$$P_n = P_a * P_b * 2$$

Pb >= 0.5の時

$$P_n = 1 - 2 * (1 - P_a) * (1 - P_b)$$



※上のレイヤーは、不透明度を50%に下げています。

同じ画像同士を重ねると、明るい部分は一層明るく、暗い部分は一層暗くなるので、彩度が上がります。

8. 覆い焼き(Dodge)

覆い焼きは、ハイライトなどで頻繁に使われます。みつみ美里さん辺りがエヴァンジェリストでしょう。劇的に明るくなる上、彩度が大きく変わりますので、瞳や髪、肌のハイライトにちょっとだけ使うのがポイントです。金属らしさを表現する際にも使えます。

この関数は、ゼロ除算に注意してください。というか、露骨なゼロ除算だなあ。

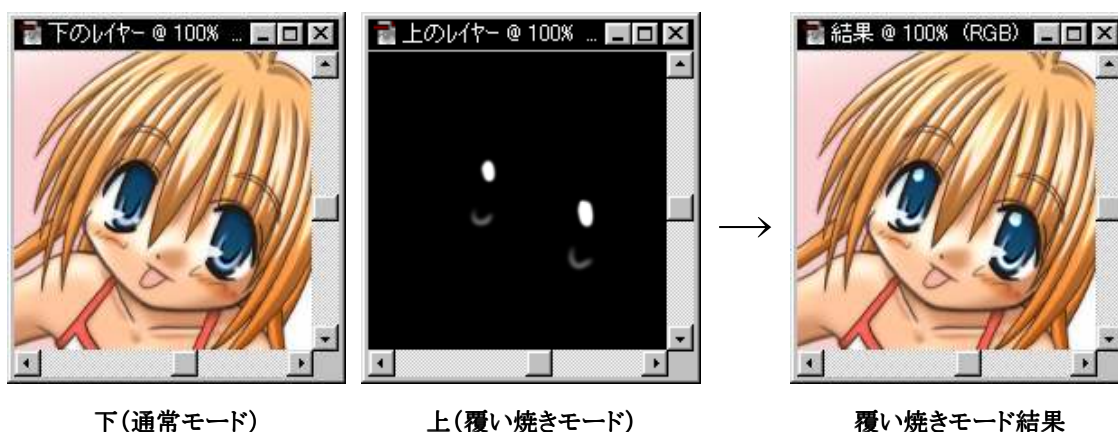
$$P_n = P_a * 255 / (255 - P_b)$$

ただし、Pb = 255 の場合及び Pn > 255 の場合、Pn = 255とする(こういうのを“saturate”というらしい)

Pn、Pa、Pb を0～1の範囲とする場合、

$$P_n = P_a / (1 - P_b)$$

PhotoShop6.0 では挙動が変わっています。6.0の関数は調べてないので知りません。レイヤーセットを使用すると、上記と同じ効果が得られます。



※上のレイヤーは瞳の塗り部分とクリッピンググループ化してあります。



※上のレイヤーはパイプの塗り部分とクリッピンググループ化してあります。

9. 焼き込み(Burn)

焼き込みは金属っぽさを表現できます。塗り斑に使うこともできます。ブラシを焼き込みモードにして陰(shade)部分を塗る人もいましたが、鮮やかになりすぎて金属っぽくなるので、そう意図していない限り止めた方が無難です。

計算式は以下の通りです。これもゼロ除算に注意してください。

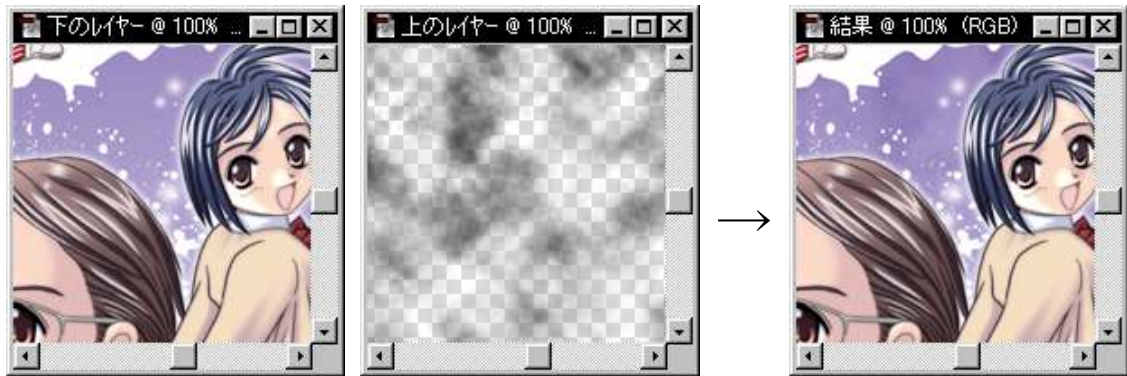
$$P_n = 255 - ((255 - P_a) * 255 / P_b)$$

ただし、 $P_b = 0$ の場合及び $P_n < 0$ の場合、 $P_n = 0$

P_n 、 P_a 、 P_b を0～1の範囲とする場合、

$$P_n = 1 - (1 - P_a) / P_b$$

覆い焼き同様、PhotoShop6.0 では挙動が変わっています。レイヤーセットを使用すると、上記と同じ効果が得られます。



下(通常モード)

上(焼き込みモード)

焼き込みモード結果

※上のレイヤーは人物等の後ろ、背景の手前で、不透明度を30%に下げています。

10. 比較(暗)(Darken)

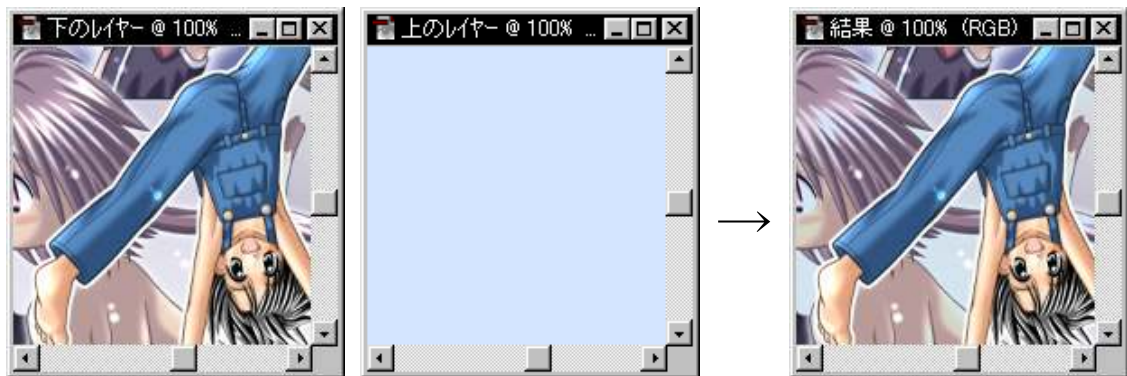
これはとても簡単。RGB 値のローパスフィルタと言っても好いかもかもしれません。上のレイヤーの RGB 値でカットオフします。

$P_a \geq P_b$ の時

$P_n = P_b$

$P_a < P_b$ の時

$P_n = P_a$



下(通常モード)

上(比較(暗))モード

比較(暗)モード結果

※上のレイヤーは、背景の上、手前の人物の下に挿入しています。

特定の値以上を急激にカットしますから、ヒストグラムはイマイチな形になります。明度や輝度をコントロールするのが目的ならば、レベル補正を使うべきです。わざと画像を劣化させたり、RGB の特定チャンネルをカットする際には使えるかもしれませんが。背景のコントラストを落としたりとか。

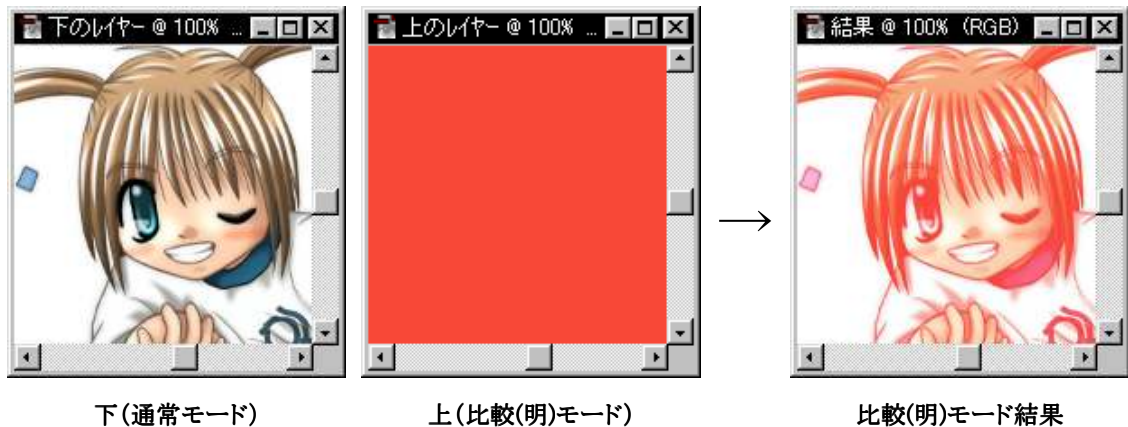
11. 比較(明)(Lighten)

RGB 値のハイパスフィルタ(なんて呼ぶとお叱りが来そうだが……)です。

$P_a \geq P_b$ の時

$P_n = P_a$
 $P_a < P_b$ の時
 $P_n = P_b$

これもチャンネルの情報量を削ってヒストグラムを潰しますから、明度や輝度をコントロールするのが目的ならばレベル補正を使うべきです。



こういう風に画像を劣化させるのも、たまになら面白いですね。

12. 差の絶対値(difference)

下のレイヤーと同じ画像を重ねると真っ黒になるため、2枚の画像の差異を調べたり位置を合わせたりする際に利用できます。また、グレースケール画像の場合、ベタにかかる効果線などを自動的に白黒反転させたりすることができます。

値を求める関数自体は名前通りなので大変容易です。

$P_n = | P_a - P_b |$

RGB 値を反転させる時にも使えそうな気もするのですが、エッジ部分が汚くなるので、私はあまり好きではありません。また、反転した RGB 値は、輝度などの問題から視覚的な補色としてふさわしくないことが多い上に細かい調整がし難いので、色相や明度を調整して通常モードで乗せる方がずっと楽でしょう。

13. 除外(exclusion)

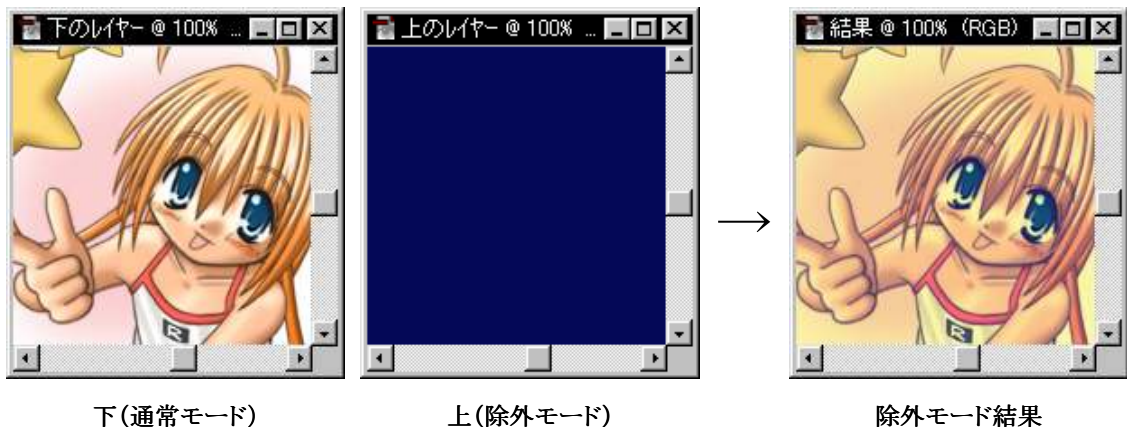
差の絶対値と似てるようで、実は違います。元の値とビット反転した値との差をビット深度で割ったものが傾きになる関数のようです。私は実際の作業で使ったことがありません。

計算式は以下の通り。

$$P_n = P_a + P_b - 2 * P_a * P_b / 255$$

P_n 、 P_a 、 P_b を0~1の範囲とする場合は、

$$P_n = (1 - P_b) * P_a + (1 - P_a) * P_b$$



※差の絶対値よりヒストグラムを潰すので柔らかく見える。



参考: 差の絶対値での結果

14. ここまでのまとめ

ソフトライトの項でも触れましたが、以上のブレンドモードは、RGB 各チャンネルに対して平等に働きます。これが、人間の感覚的な明るさの認識(心理物理量)と一致しないため、いまいち直感的ではないものもあると思います。例えば 覆い焼き では、色によっては狙ったように明るくならないことがあります。これは残念ながら慣れるしかありません。

2. 色相と輝度を利用したブレンド

色相と輝度を利用したブレンドモードには以下の 4 つがあります。

1. 色相(Hue)

2. 彩度(Saturation)
3. カラー(色相と彩度)(Color)
4. 輝度(Luminance / Luminosity)

このブレンドモードは、基本的には HSB カラーモデル を使用していますが、普通の HSB モデルとは異なり、明度 (Brightness) ではなく輝度 (Luminance) に大きく支配されています。というよりも、明度は無視しています。ですから、これを HSB モデルと呼ぶのは問題があるかもしれません。輝度とは、

$$\text{Luminance} = (0.298912 * r + 0.586611 * g + 0.114478 * b);$$

で計算される明るさの指標です。HSB の明度は RGB の最大値から算術的に計算されます が、輝度は人間の視覚に合わせた感覚的な指標です。上記の係数では細かすぎるので、それぞれ 0.30、0.59、0.11 程度で充分でしょう。テレビの Y / C 分離や JPEG 内部の YUV でも、同じ理由から輝度を分離して処理します。輝度と色相または彩度だけで色を決めるというのは、お互いが依存しあってるので結構厄介です。

1. 色相(Hue)

上のレイヤーの色相で下のレイヤーを着色しますが、下のレイヤーの輝度を維持します。つまり、上のレイヤーの彩度や明度は一切維持されませんので要注意です。

ちょっと計算式を上手くまとめられなのですが、ルールは以下の通りです。

上のレイヤーの色相を H_b 、下のレイヤーの輝度を L_a 、結果色の色相を H_n 、下のレイヤーの RGB の最小値と最大値の差を C_a とすると、最終的に合成される RGB 値は、以下の2式

$$H_n = H_b$$

$$L_a = (0.30 * r + 0.59 * g + 0.11 * b)$$

を満たす r 、 g 、 b の組み合わせのうち、その最大値と最小値の差 (C_n とする) が C_a と等しいものである。ただし、 C_a が C_n の最大値より大きい場合は、 C_n が最大になるものとする。

例を出すと、

204 110 51 ■

に

65 188 106 ■

を色相で重ねる場合、

$$L_a = 0.30 * 204 + 110 * 0.59 + 0.11 * 51 = 132$$

$$H_b = H_n = 140$$

を満たす組み合わせは、

131 133 132 ■

130 133 131 ■

128 135 131 ■

<中略>

12 204 76 ■

8 207 74 ■

6 207 73 ■

4 209 72 ■

0 212 71 ■

など多数あるが、

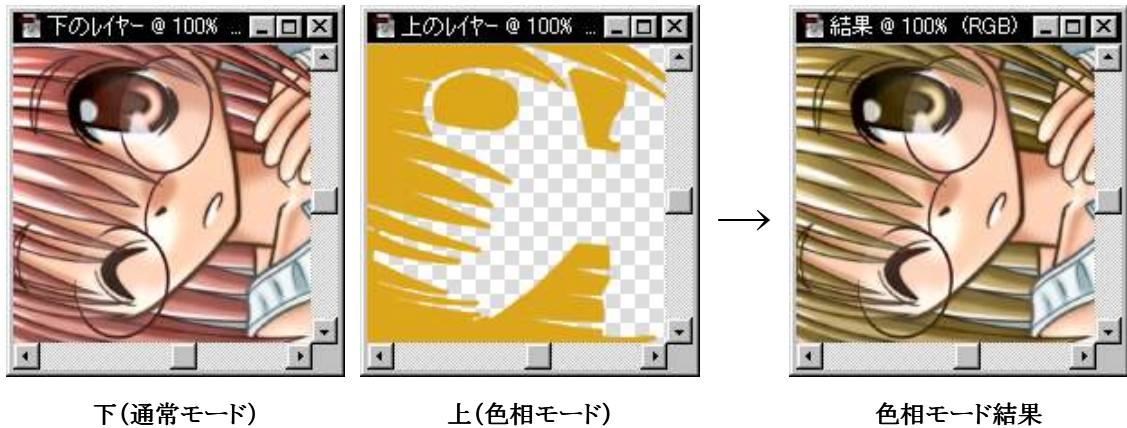
Ca=153

なので、Cn=153 となるような

36 189 87 ■

が結果となる。

ルールとしては上記の通りなのですが、この考え方で実装したら、たかだかブレンドの表示のためにループか再帰が必要になるのでちょっとヤバそうです。YCrCb で考えても結局同じだと思うし……。うーん実際にはどうやってんだらう。私が式を思いつけてないだけだと思うんですが。



調整レイヤーで色補正するよりも、このように色相レイヤーで補正することをお勧めします。

2. 彩度(Saturation)

これも下のレイヤーの輝度を維持しようとするので、上のレイヤーの彩度は完全には維持されません。正確に書くと、上のレイヤーの色相と輝度を下のレイヤーのものに置き換えます。

ルールは以下の通り。

下のレイヤーの色相を H_a 、下のレイヤーの輝度を L_a 、結果色の色相を H_n 、上のレイヤーの RGB の最小値と最大値の差を C_b とすると、最終的に合成される RGB 値は、以下の2式

$$H_n = H_a$$

$$L_a = (0.30 * r + 0.59 * g + 0.11 * b)$$

を満たす r 、 g 、 b の組み合わせのうち、その最大値と最小値の差 (C_n とする) が C_b と等しいものである。ただし、 C_b が C_n の最大値より大きい場合は、 C_n が最大になるものとする。

→



下(通常モード)



上(彩度モード)



彩度モード結果

※上のレイヤーは不透明度を 80% にしてあります。

彩度による塗り斑も結構面白い効果があるかと。

3. カラー(色相と彩度)(Color)

これも、ヘルプの記述はやや不正確です。上のレイヤーの色相は維持されますが、彩度は輝度によって変化します。元画像の輝度が飛んでしまうので、ベタ塗りのレイヤーで色調を統一するためだけなら、これを使わずに彩度モードを使いましょう。逆にいえば、輝度を飛ばして均すことができます。

ルールは以下の通り。

上のレイヤーの色相を H_b 、下のレイヤーの輝度を L_a 、結果色の色相を H_n 、上のレイヤーの RGB の最小値と最大値の差を C_b とすると、最終的に合成される RGB 値は、以下の2式

$$H_n = H_b$$

$$L_a = (0.30 * r + 0.59 * g + 0.11 * b)$$

を満たす r 、 g 、 b の組み合わせのうち、その最大値と最小値の差 (C_n とする) が C_b になるものである。

これは彩度モードでも同じことですが、白や黒の無彩色の色を乗せると、グレースケール化することができます。輝度を維持したまま下のレイヤーの彩度を落とすなどの際に使えると思います。

4. 輝度(Luminance / Luminosity)

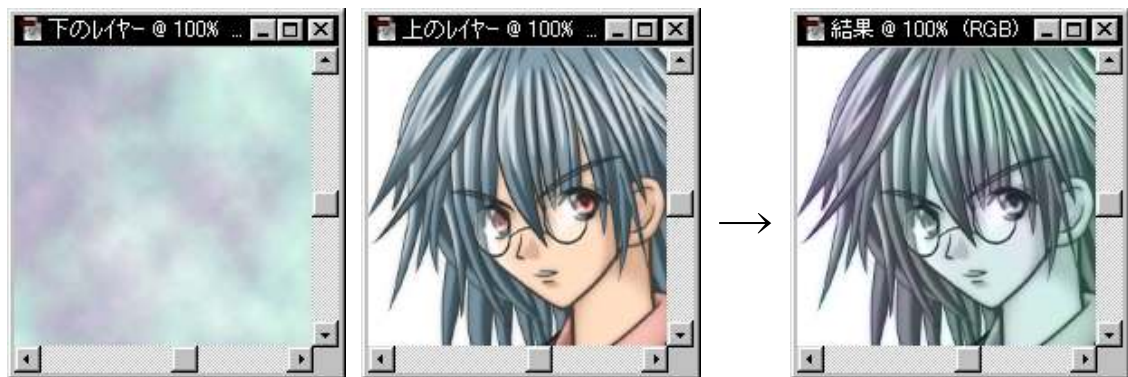
上のレイヤーの輝度で下のレイヤーを均します。輝度を均すと画像のイメージが大幅に変わるので要注意です。

下のレイヤーの色相を H_a 、下のレイヤーの輝度を L_a 、結果色の色相を H_n 、下のレイヤーの RGB の最小値と最大値の差を C_a とすると、最終的に合成される RGB 値は、以下の2式

$$H_n = H_a$$

$$L_n = (0.30 * r + 0.59 * g + 0.11 * b)$$

を満たす r 、 g 、 b の組み合わせのうち、その最大値と最小値の差 (C_n とする) が C_a と等しいものである。



下(通常モード)

上(輝度モード)

輝度モード結果

これはなかなか使い道思いつきませんね……。輝度は画像の一番重要な情報なので均すわけにはいかないし。わざわざ輝度を潰さねばならんシチュエーションって何だろ。

3. まとめ

幾つか思いつく注意点だけをまとめておきます。

1. レイヤーモードでやるよりは、画像をコピーして調整して通常モードで重ねた方が応用が効くことも多いので、無理にレイヤーモードを使わない方が生産性上がるかも。
2. RGB モードのブレンドの一部には、直感的な予想と結果とが異なるケースがあるが、諦めて慣れるしかない。
3. 全体の色を調整する場合は、調整レイヤーの 色相・彩度 ではなく、色相モードのレイヤーを使う方が、輝度が維持されるのでお勧めである。