

7 色と画像の表現

7.1 色とは

色の感覚を引き起こす光，すなわち，人が目で見ることができる可視光は電磁波の一種です。電磁波とは，電場と磁場が振動することによって生じる波です。例えば，携帯電話やテレビの電波も電磁波の一種です。またX線やガンマ線なども電磁波の一種です。図7-1では，おおまかに，電磁波の波長と種類の関係を示しています。電磁波の中で波長が780nmから380nmの範囲の電磁波は目で知覚することができます。この範囲の電磁波は可視光と呼ばれます。可視光は，波長が長い方から赤，橙，黄，緑，青，藍，紫に対応します。

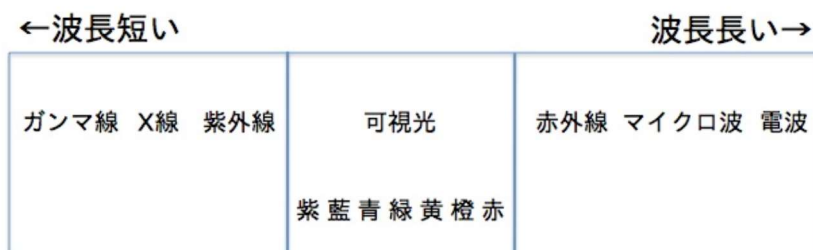


図7-1 電磁波

では，このような電磁波を人はどのように知覚しているのでしょうか。人が物を見る場合，レンズの働きをする水晶体によって光が網膜上に集められます。網膜上には，明るさを感じるための杆体と呼ばれるセンサーが存在します。また，色を感じるための錐体と呼ばれるセンサーが存在します。杆体は色を区別できないものの，錐体よりも弱い光を捉えることができます。暗いところで色が見えにくくなるのは，主に杆体により光を捉えているためです。錐体には赤い光に反応する錐体，緑の光に反応する錐体，青い光に反応する錐体の3つがあります。人は，この3つの錐体からの信号の大小により色を感じています。このため，3つの錐体からの出力が同じであれば同じ色と感じます。

逆に，色を合成することを考えてみましょう。人は，3つの錐体からの出力が同じであれば同じ色とを感じるわけですから，赤，緑，青の3色を組み合わせることで任意の色を作り出すことができます。これが光の3原色です。図7-2は，光の3原色を示したものです。光の3原色である，赤，緑，青と，それらを重ね合わせた場合の色が示されています。赤と緑を足し合わせることで黄色（イエロー）になります。緑と青を足し合わせることでシアンになります。青と赤を足し合わせることでマゼンタになります。そして，赤と緑と青を足し合わせることで白を作り出すことができます。

ここで注意すべきことは，黄色と青を足すと白を作り出せることです。黄色は，赤と緑を足し合わせたものです。このため，黄色+青は，赤+緑+青となり，これは白となります。黄色と青のように，2つの色を合わせることで白を作り出せる色は互いに補色の関係になります。光の3原色の図で，黄色と青は，白を挟んで反対側に位置しています。このように，白を挟んで反対側に位置する色は，補色になります。例えば，赤はシアンの補色になります。また，緑はマゼンタの補色になります。光の3原色は，色を重ねれば重ねるほど明るくなるという性質があります。このような混色は加法混色と呼ばれます。

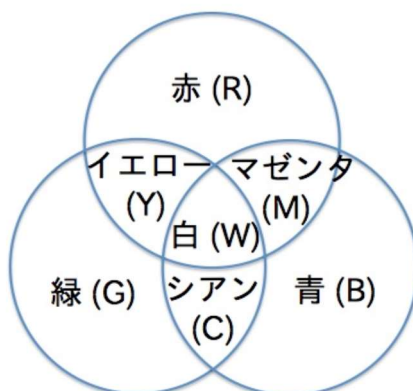


図7-2 光の3原色

次に、カラーで印刷する場合を考えてみましょう。光の3原色は、赤、緑、青であることを述べました。しかしながら、印刷用のインクは、赤、緑、青ではありません。これは、なぜでしょうか。印刷用のインクが、赤、緑、青ではないのは、印刷用のインクは自分では発光しないためです。では、どうやって色を作り出すかと言えば、印刷の場合、白色光から不要な色成分を取り除き、印刷物からの反射光が必要な色になるようにします。すなわち、印刷物の場合、太陽光など、外から光を当てることが必須になります。そして、例えば、赤を吸収するインクでは、白色光から赤の成分を取り除くため、反射した光は、赤の補色のシアンに見えます。同様に、緑を吸収するインクでは、白色光から緑の成分を取り除くため、反射した光は、緑の補色のマゼンタに見えます。また、青を吸収するインクでは、白色光から青の成分を取り除くため、反射した光は、青の補色のイエローに見えます。

このように、印刷用のインクとして用いられる色の3原色は、シアン、マゼンタ、イエローの3色になります。図7-3では、色の3原色である、シアン、マゼンタ、イエローと、それらを重ね合わせた場合の色を示しています。例えば、シアンのインクとマゼンタのインクを混ぜ合わせることで、青を作り出すことができます。これは、シアンが赤の色成分を吸収し、マゼンタが緑の色成分を吸収するため、青の色成分のみが反射光に残るためです。シアンとマゼンタとイエローのインクを混ぜ合わせることで、黒を作り出すことができます。シアンが赤の色成分を吸収し、マゼンタが緑の色成分を吸収し、イエローが青の色成分を吸収することで、すべての光を吸収する黒を作ることができます。この色の3原色は、光の3原色とは補色の関係になっています。また、印刷用のインクの場合、色を重ねれば重ねるほど暗くなりますが、このような混色は減法混色と呼ばれます。

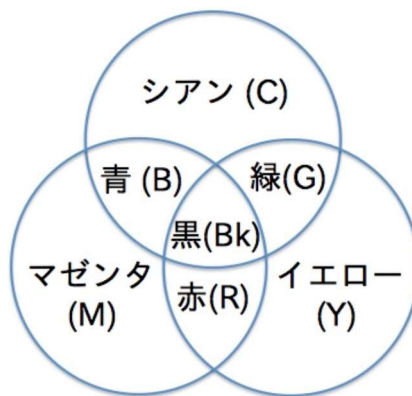


図7-3 色の3原色

7.2 色の表現

コンピュータ上では、光の3原色である、赤、緑、青の組み合わせで、任意の色を表現しています。そして、赤、緑、青の強度は、数値データとして扱われます。具体的には、コンピュータ上で色を表す場合、通常、赤 (R)、緑 (G)、青 (B) の強度をそれぞれ1バイトで表現します。すなわち、RGBのそれぞれの強度を、1バイトで表現できる、0から255の256段階の数値で表現します。1つの色を表現するためには、RGBそれぞれに1バイト必要になるため、全体で3バイトのデータが必要となります。フルカラーのコンピュータでは、この3バイトで表現できる $256 \times 256 \times 256$ 色、すなわち、約1600万色を表現することができます。

具体的な色の表現は、RGBの強度を3つ並べたベクトルの形で表現することができます。例えば、赤は、Rの強度が最大の255になり、G、Bの値は0になります。このため、RGBのベクトルの形で表現すれば、(255,0,0)と表現することができます。同様に、緑は、Gの値が最大の255で、RとBが0になることから(0,255,0)と表現できます。青は、Bの値が最大の255で、RとGが0になることから(0,0,255)と表現できます。

次に、2つの色を混ぜた場合を考えてみましょう。黄色は、赤と緑により作られるため、RとGが255、Bが0で、(255,255,0)と表現できます。シアンは、緑と青により作られるため、GとBが255、Rが0で、(0,255,255)と表現できます。マゼンタは、赤と青により作られるため、RとBが255、Gが0で、(255,0,255)と表現することができます。

次に無彩色の色を見てみましょう。無彩色とは、白や黒といった色味を持たない色です。RGBの値がすべて同じになると無彩色になります。白は、RGBの値がすべて最大値の255となり、(255,255,255)と表現できます。黒は、RGBすべての光の強度が0となるため、(0,0,0)と表現することができます。白と黒の間の灰色は、例えば、(128,128,128)のように表現することができます。

では、暗い赤や明るい赤はどのように表現できるでしょうか。暗い赤は、(255,0,0)のRの値を255より小さくすることで作り出すことができます。このため、暗い赤は、例えば、(128,0,0)のように表現することができます。明るい赤は、(255,0,0)のGやBの値を大きくすることで作り出すことができます。例えば、明るい赤は、(255,128,128)のように表現することができます。

補色については、赤とシアンを足し合わせると、白になることをお話ししました。コンピュータ上では、赤は(255,0,0)と表現され、シアンは、(0,255,255)と表現されます。これらを足し合わせると、(255,255,255)となり、白になることが分かります。このように、RGBの強度を数値化することで、コンピュータ上の色は表現されています。

ここで、人の感じる色について、もう少し詳しく見てみましょう。人の目は、色を赤、緑、青の強度として知覚することをお話ししました。しかし、人が実際に色を感じる場合は、赤、緑、青の強度として感じるのではなく、色相、明度、彩度として色を感じます。美術の教科書などに出てくるマンセル表色系は、このような、色相、明度、彩度の空間の中に実際の色を配置したものになります。色を円筒形の空

間で表現した場合、図7-4のようになります。明度は、明るさを表します。この図で、明度は、上下方向のZ軸に対応します。Z軸方向の上に行くほど明るい色になり、下に行くほど暗い色になります。色相は色合いを表します。色相は、X軸との角度に対応します。X軸との角度が大きくなるにつれて、赤、黄、緑、シアン、青、マゼンタと変化します。彩度は色の鮮やかさを表します。この図では、Z軸からの距離に対応します。Z軸からの距離が大きくなるほど色が鮮やかになります。例えば、純粋な赤、緑、青は、円筒の軸から最も遠いところに並びます。逆に、円筒の中心であるZ軸上には、下から黒、灰色、白といった無彩色が並びます。

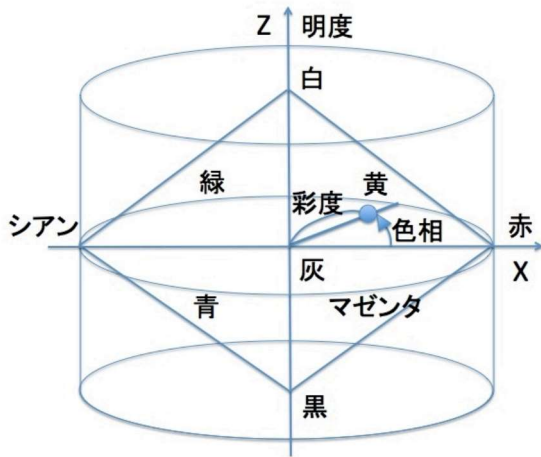


図7-4 色相/明度/彩度

RGBから色相、明度、彩度への変換は人間の感覚と関係するため正確に変換することは必ずしも容易ではありません。このため、変換の容易さを優先した方法や、変換の正確さを優先する方法など、さまざまな変換方法が提案されています。例えば、HSB色空間は、コンピュータ上で簡易に色相、明度、彩度を計算することができるものです。HSVと呼ばれる場合もあります。また、国際照明委員会により提案された $L^*a^*b^*$ や $L^*u^*v^*$ といった変換方式は、より複雑な変換式を用いることで、より人間の感覚に近い色相、明度、彩度への変換を実現することができます。 $L^*a^*b^*$ や $L^*u^*v^*$ は、色の違いが心理的に同じになる2つの色の間の距離が均等になるように、色空間を形づくりします。コンピュータ上で色情報を扱う場合、RGBの値を直接用いる方が一般に処理は容易になります。しかし、人の感覚により近い処理を行うためには、RGBで表現されている色を、色相、明度、彩度へと変換して処理することが必要になります。

7.3 画像の表現

図7-5は画像の例ですが、一部を拡大すると、画像は点で構成されていることが分かります。これらの点は、それぞれ色を表しており、これらの点を2次元に並べることで画像を表現することができます。画像を構成する最小単位は画素、あるいはピクセルと呼ばれます。

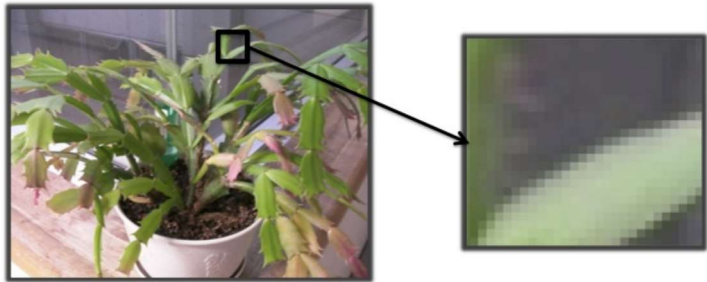


図7-5 画像と画素

一般に、文字などに比べて画像はデータ量が大きくなります。例えば、横640画素、縦480画素の画像を考えます。1つの色を表すためには、RGBそれぞれ1バイトずつで、3バイト必要になります。横640画素、縦480画素の画像であれば、 $640 \times 480 \times 3$ で、約900kBとなります。画像と文字のデータ量を比較すると、日本語の文字は通常1文字2バイトで表現されます。このため、 640×480 の画像は、 $640 \times 480 \times 3/2$ で、およそ45万文字分のデータ量になります。このように画像のデータ量は大きく、そのまま扱うことは効率的ではありません。このため、画像を保存する際には、通常、データ量を小さくするための圧縮と呼ばれる処理が行われます。

圧縮方法の1つとして「ランレングス符号化」があります。例えば、画像の例として、白と黒が

白白白白白白
白白黒黒白白
白白白白黒白白
白白白白白白

と並んだデータがあったとします。このデータは、白と黒の数を交互に書くことで、

のように表現することもできます。この場合、もとの画像を完全に復元できますが、このような圧縮は、可逆圧縮と呼ばれます。

一方、カラー画像では、色数を落とすことで、データ量を削減することもできます。例えば、フルカラー（約1600万色）の画像の色数を256色といった、より少ない色数に落とします。フルカラーの画像では、RGBそれぞれ1バイト、すなわち3バイトで1つの色を表現しています。色数を256色に落とした場合、それぞれの色は、0番の色、255番の色というように、1つの色を1バイトで表現できます。それぞれの色番号に対するRGBの値は、別途用意する必要がありますが、もともと3バイトで色を表現していたものが、1バイトで表現できるようになります。このため、画像のデータ量を1/3程度に圧縮することができます。しかし、もとの画像の色数が256色を超える場合は、圧縮により、色が変わる画素が出てくるため、もとの画像を完全には復元できなくなります。

このような画像データを保存するための形式（フォーマット）としては、GIF、PNG、JPEGといったさまざまなフォーマットがあります。GIFは、256色以下の画像を保存するためのフォーマットです。1バイトの色番号と、それぞれの色番号に対応するRGBの値などからデータが構成されています。可逆圧縮のフォーマットであり、256色以下の画像では、圧縮によって画質の劣化がありません。PNGは、GIFのように、色数が256色以下といった制限が無く、フルカラーの画像を扱うことができます。また、可逆圧縮のフォーマットであり、圧縮により、画像の劣化がないという利点があります。JPEGでは、画像中の細かな構造を表現するために必要な情報量を削減するなどして圧縮が行われます。圧縮により、もとの画像を完全には復元できなくなりますが、圧縮率が高いという特徴があります。

画像フォーマットの使い分けについては、例えばWebページ上にボタンの画像を貼り付けるような場合を考えてみましょう。通常、ボタンの色数は、256色以下であることが多いため、GIFなどのフォーマットが利用されます。もし、ボタンのような、特に輪郭がシャープな画像をJPEGで保存すると、輪郭近くに、もやもやとしたノイズが乗る場合があります。一方、自然の風景を撮影したフルカラーの画像を保存するような場合を考えてみましょう。このような画像をGIF形式で保存しようとする、色数を256色に落とす必要が出てきます。この際、擬似的な輪郭が生じてしまう場合があります。このため、自然の風景を撮影したような画像の場合、通常は、JPEGなどのフォーマットが利用されます。このように、目的などに応じて、どの画像フォーマットを利用するかを考える必要があります。

次に、画像の解像度についてお話しします。解像度とは、画像のきめ細やかさを表す指標です。画像を読み込むスキャナや、画像を印刷するプリンタなどの解像度は、通常1インチ当たりの画素数で表されます。単位はdpi（dot per inch）が用いられます。一般に、解像度が高いほど画像がきめ細くなり画質が良くなります。反面、解像度が高くなるほど、データ量は大きくなります。

一方、コンピュータのモニターで解像度と言う場合は、通常、画面を構成する点の数を表します。640×480の画面はVGAと呼ばれます。また、1024×768の画面はXGAと呼ばれます。また、フルHDと言う場合の画面の解像度は、1920×1080となります。

以上述べてきたような、画像を画素の集合として表現したデータはラスタデータと呼ばれます。ラスタデータは、2次元画像を描くためのペイントソフトや写真の修整等を行うフォトタッチソフトなどで用いられます。ラスタデータは画素の集合として表現されているため、拡大すると、画素が拡大されて画像が粗くなります。

一方、画像の表現形式として、ベクタデータ（ベクトルデータ）と呼ばれる形式もあります。ベクタデータは、ドローソフトなどで用いられます。ベクタデータでは、図形を特徴的な点とそれらを結ぶ線で表現します。このため、拡大や変形をしても、図形の滑らかさが保たれるという特徴があります。

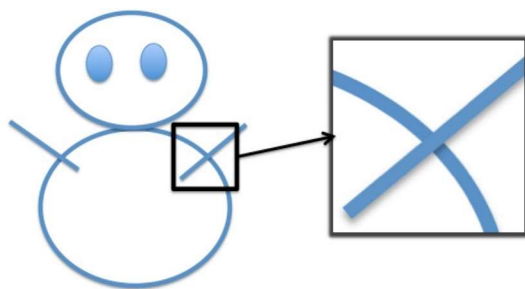


図7-6 ベクタデータ