

1 はじめに

"画像フォーマット"って知っていますか？ そんな言葉聞いたこともない人、知っているけど普段は意識していない人、各画像フォーマットの仕様まで知っている人、時代はJPEG2000だよって人、色んな人がいるんじゃないかと思います。分からないって人がいたら、自分のPCのマイピクチャでも見てみてください。そこにある画像に、"JPEG" だの "GIF" だの "PNG" だのと書いてないでしょうか？ そいつらが画像フォーマットというやつで、簡単に言うとコンピュータにおける画像の保存形式のことを指します。

現在画像フォーマットにはとても多くの種類がありますが、扱う種類としてはビットマップ画像とベクトル画像の2種類に、用途としては画面（ディスプレイ）表示用と印刷用の2種類に大別できます。がしかし、それらの画像フォーマットの特徴を理解して使い分けられている人は、それほど多くないんじゃないかと思います。実際周りの人に尋ねてみたら、10人中9人が使い分けられていなかったり、印刷に使うB5サイズの線画イラストを注文したら、72dpiのJPEGで送られてきてメルトしかけたことがあったり……なんて状況でした。

じゃあ画像フォーマットって具体的にどう使い分ければいいんでしょうか？ そして使い分けると何が嬉しいんでしょうか？ この記事ではそんな疑問に答えるために、僕の独断と偏見による一方的な画像に対する知識と、それを踏まえた上でどんな時にどんなフォーマットを使えばいいかってことを書いていきたいと思います。

2 基本的なこと

2.1 画像の種類

最初に書いたように、画像にはビットマップ画像とベクトル画像の2種類があります。ビットマップ画像（ラスタ画像とも表現）は、多分普通の人が"画像"と聞いてまず思い浮かべる種類の画像じゃないかと思います。インターネットで見かける画像も、自分のPCに保存している画像も、その大半がビットマップ画像だと断定できると言っても過言ではないくらいです。じゃあベクトル画像（ベクター画像とも表現）って何でしょうか？ その名の通り"ベクトル"を使った画像なんですけど、大抵の人はあまり触れる機会がないと思います。

とりあえず2つの画像の特徴を挙げてみるので、両者の違いを見てみましょう。

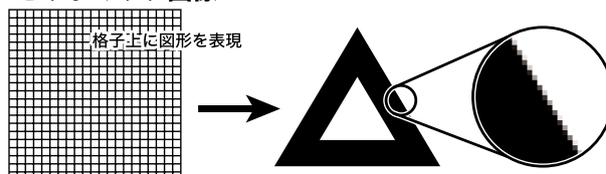
ビットマップ画像

- ・ 格子状に並んだ点の集合で画像を表現
- ・ 画質は解像度で決まる
- ・ 拡大 / 縮小 / 回転処理によって画質が低下する

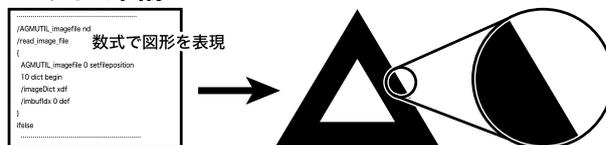
ベクトル画像

- ・ 数式を元に画像を図形として表現
- ・ 解像度という概念がない
- ・ 拡大 / 縮小 / 回転処理をしても画質が変化しない

ビットマップ画像



ベクトル画像



これだけ見たら「ベクトル画像の方がいいことづくめじゃねーか」と思うかもしれませんが、実際は世の中そんなに甘くありません。例えばベクトル画像は、ぼかしやグラデーションのような、画像に複雑な変化を持たせる処理があんまり得意じゃありません。そういう処理をするなら、ビットマップ画像の方が断然有利なんです。またベクトル画像は、あくまで数式を元にして描画する（といってももちろん実際に数式を打ち込んで操作するわけじゃないです）ので、直感的な操作をすることができず、初めて操作をする時はかなり戸惑うと思います。その点ビットマップ画像は、紙に絵を描くのと同じ感覚で絵を描くことができます。

じゃあどういう時にどっちの画像を使えばいいんでしょうか？ 複雑に色が変化する画像、つまり写真のような画像には、ビットマップ画像が適しています。輪郭線がはっきりくっきりしている・拡大縮小が想定されるなどの画像、つまり会社や学校なんかのロゴみたいなイラストや文字には、ベクトル画像が適しています。また画像フォーマットには、ビットマップデータだけ含めるもの、ベクトルデータだけ含めるものだけでなく、ビットマップ・ベクトルの両方を含めるものもあるので、「写真とロゴを合わせた画像を作りたい」なんて時でもノープロブレムです。

また、ビットマップデータをベクトルデータに変換（ベクタライズ）、ベクトルデータをビットマップデータに変換（ラスタライズ）、なんて処理も可能です。ラスタライズは「ベクトルデータが見た目そのままビットマップデータになる」って認識でいいんですが、ベクタライズではそんなに正確に変換できません。数式から点の集合を計算するのは簡単でも、点の集合から数式を計

算するのは難しいからです。

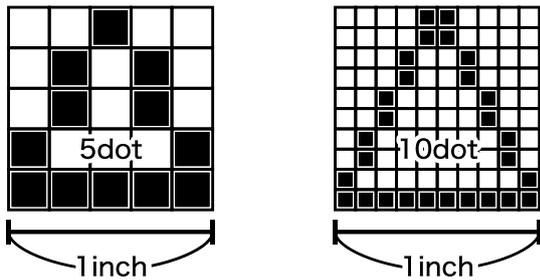
基本的にどんな画像でもビットマップ・ベクトルの一方だけで表せますが、それぞれの長所・短所を考慮しながら、ビットマップ・ベクトルのどちらで表現するかを選びましょう。

画像にはビットマップ画像とベクトル画像の2種類がある。2つの画像は得意な分野が異なるので、用途に合わせて使い分けることが大切。

2.2 解像度

ビットマップ画像とベクトル画像の特徴を比較した際に、“解像度”なる単語が出てきました。解像度とは、単位長さあたりにどれだけの画素が含まれるかを表す指標で、前述のようにビットマップ画像にしかない概念です。

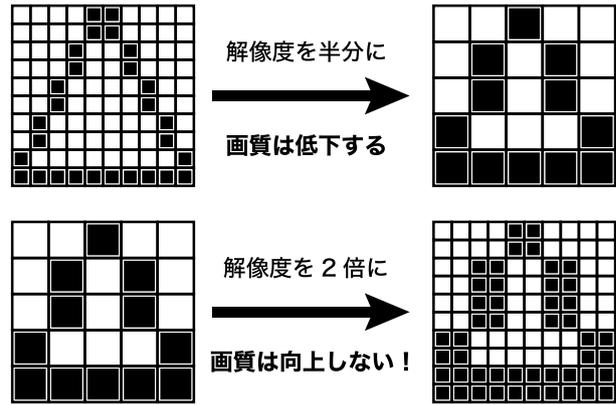
解像度の単位には、通常 dpi もしくは ppi (Dot Per Inch / Pixel Per Inch : 1 インチ当たりの画素数) が用いられます。Dot も Pixel も同じなので、つまるところ2つの単位は同じものと考えて差し支えありません。基本的に dpi はディスプレイや印刷に出力するものの解像度に対して、ppi はディスプレイに出力するものの解像度に対して使われると覚えておけば大丈夫です。普段は dpi を使えば問題ないでしょう。



$5[\text{dot}] / 1[\text{inch}] = 5[\text{dpi}]$ $10[\text{dot}] / 1[\text{inch}] = 10[\text{dpi}]$

解像度が高ければきめの細かい画像になって画質も向上しますが、もちろんそうなるにつれて格子の数が増えるので、画像のファイルサイズも増加します。ただ気を付けないといけないのは、解像度を下げるとそれに比例して画質も下がりますが、解像度を上げて画質は向上しないってことです。何故なら小さい解像度から大きい解像度に変化させても、大きな点を小さな点の集合で表し直すだけだからです。おまけに画質は変わらないのに、ファイルサイズだけが大きくなって踏んだり蹴ったり。これは逆に言えば、例えば最終的に 100dpi で出力をするつもりなら、画像の編集行程において1度も 100dpi を下回ってはいけないってことです。常に最終的に必要な解像度以上を維持しながら編集することが要求されます。

解像度は高すぎるとファイルサイズが無駄に大きくなり、低すぎると粗悪な画像になってしまいます。最適な解像度を考えて設定しましょう。



ディスプレイにしか出力しない場合には、画像の解像度は特に気にする必要はありません。何故ならディスプレイ自体の解像度が決まっているので、画像自体の解像度を変更しても、ディスプレイで見る限りは何も変わらないからです。しかしプリンタで出力する場合は、画像の解像度をよく考えて設定する必要があります。

例えば解像度が 100dpi のディスプレイと、1000 ドット × 500 ドットの画像があるとします。まず画像自体の解像度が 50dpi の場合、ディスプレイでは 10 インチ × 5 インチの大ききで表示され、プリンタでは 20 インチ × 10 インチの大ききで出力されます。一方画像の解像度を 250dpi にすると、ディスプレイでは同じく 10 インチ × 5 インチの大ききで表示されますが、プリンタでは 4 インチ × 2 インチの大ききで出力されます。要するにディスプレイの場合はドットの数だけで大ききが決まるってことです (ディスプレイの解像度を変えたら表示される大ききも変わります)。

このように、印刷をする場合だけ解像度を考慮していれば問題ありません。じゃあその場合には解像度をどのくらいに設定していればいいのでしょうか? 印刷の仕方にもよりけりですが、大体 300 ~ 400dpi の解像度があれば、線がガタガタにならず、綺麗に印刷できると言われています。ちなみに解像度を何も設定しないで画像を作成すると、大抵の場合は勝手に 72dpi に設定されます。つまり解像度を気にせずに印刷したら、思わずメルトする低クオリティな印刷結果になることが予想できるでしょう。最終的に印刷をするつもりなら、解像度に気をつけて画像を作成することをおすすめします。

印刷をするつもりがないなら解像度は気にしなくてよい。印刷をするつもりなら解像度を大きめに設定する。

2.3 圧縮方式

画像はテキストなどと比べるとデータ量が半端なく多いので、データを生 (= 無圧縮) のまま扱うとファイルサイズがとてつもなく大きくなってしまいます。もちろんデータを無圧縮で保存する画像フォーマットもありますが、インターネットで公開したり、他人とやりとりしたりするような用途には、転送量が多くなってしまいうのであまり向いていません。そのためそういった用途の画像は、

色んな方法でファイルサイズを減らす工夫 (= 圧縮) がなされます。

画像に対する圧縮っていうのは、基本的にビットマップ画像に対するものを指します。ベクトル画像はあくまで数式、つまり文字データなので、往々にしてビットマップ画像よりもファイルサイズが小さくなります。それ故圧縮する必要がないのか、それとも何かしらの圧縮をしているのかはよく知らないんですが、とにかく僕は全く知らなくてもいい問題のようです。

圧縮の方法は大きく分けて可逆圧縮・不可逆圧縮 (非可逆圧縮とも表現) の 2 種類があるので、以下にそれぞれの特性をまとめてみます。

可逆圧縮

- ・ 元画像と比較して画像が劣化しない
- ・ データを数学的にまとめてファイルサイズを下げる
- ・ ファイルサイズはあまり小さくならない

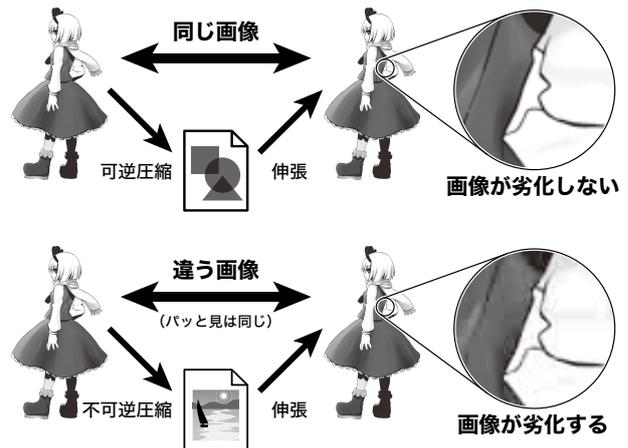
不可逆圧縮

- ・ 元画像と比較すると画像が劣化する
- ・ 人間が気付きにくい部分のデータを捨ててファイルサイズを下げる
- ・ 画質とファイルサイズは反比例する
- ・ 基本的に可逆圧縮と比較してファイルサイズはかなり小さくなる

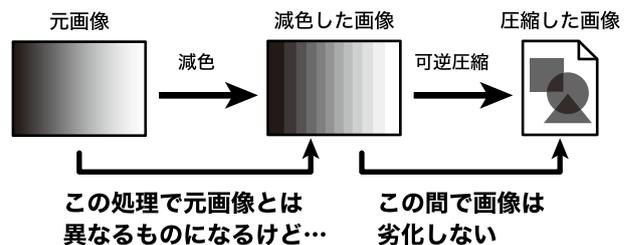
可逆圧縮をした画像は、圧縮したデータを元に戻す (= 伸張 / 展開) ことで元画像と同じ画像が得られます。一方不可逆圧縮をした画像は、元画像のデータを何かしら捨ててしまっているので、データを伸張しても元画像と同じ画像は得られません。

ここで重要なのは、不可逆圧縮した画像を可逆圧縮する画像に変換しても、画質は向上しないってことです。不可逆圧縮は、前述のように元画像のデータを一部捨ててしまっているので、一旦不可逆圧縮を経てから可逆圧縮にしても、元画像が得られないのは理解できると思います。むしろ画質は変化しないのに、ファイルサイズだけ大きくなってしまいます。

そしてもう 1 つ重要なことがあって、不可逆圧縮は圧縮をする度に画質が下がっていきます。例として、1 度不可逆圧縮をかけると画質は 0.8 倍 (0.8 というのは仮の値) になると表現すれば分かりやすいでしょうか。この場合、元画像の画質が 100 だとして、1 度不可逆圧縮を経ると画質は 80 に、もう 1 度経ると 64 に……という具合に、何度も不可逆圧縮をかけるとそれだけ画質が下がってしまうのです (あくまで例えなので実際はそんな顕著には下がりません)。なので、「最終的に不可逆圧縮をかけたいけど、途中で何度も保存を繰り返す」って場合には、最終的な出力だけ不可逆圧縮をするようにして、作業の途中は無圧縮もしくは可逆圧縮で保存することが重要です。



可逆圧縮には、画像に使う色数を減らしてファイルサイズを減らすものもあります。このアプローチを減色といいます。コンピュータは 1600 万以上の色を表現することができるんですが、人間の目では多少の色の違いを判断することはできません。そこで、人間が気付かないように画像の色数を減らすことで、ファイルサイズを減らすわけです。減色をすれば当然元画像とは違う画像になるんですが、これは不可逆圧縮とは違います。何故なら「元画像を減色したものを可逆圧縮する」ので、「圧縮した画像」を伸張すれば「元画像を減色したもの」と同じ画像が得られるからです。



可逆圧縮と不可逆圧縮にはそれぞれ得手・不得手があります。例えば不可逆圧縮は、色が複雑に変化する画像 (写真など) に対しては、画像の劣化があまり目立たない上にファイルサイズも抑えられるので、かなり有効です。しかし逆に、同じ色で塗り潰された面積が大きい画像 (アニメ絵や線画などの CG) に対しては、画像の劣化が目立ちやすいので、なるべく可逆圧縮を使うのがベターです。もちろん前者のような画像でも、「印刷用の大きなポスターに使う画像だから、なるべく画像を劣化させたくない」って場合なら、ファイルサイズが大きくなる代わりに画像が劣化しない可逆圧縮を使うべきですし、後者のような画像でも、「ホームページで公開する画像だから、なるべくファイルサイズを 500KB 以下に抑えたい」って場合には、いくら画像の劣化が目立ってしまうとはいえ不可逆圧縮を使うべきでしょう。



色が複雑に変化する画像には
不可逆圧縮が適している (写真など)



同じ色で塗り潰された面積が大きい画像には
可逆圧縮が適している (CG など)

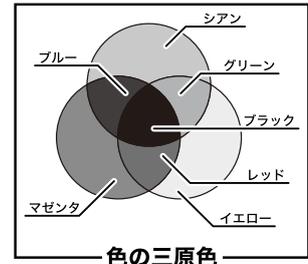
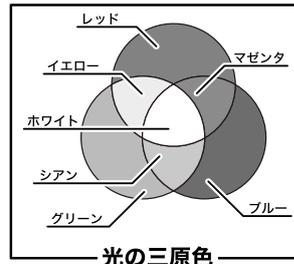
可逆圧縮はファイルサイズが大きくなるが画像は劣化しない。不可逆圧縮は画像が劣化するがファイルサイズは小さくなる。用途に合わせて圧縮方法を選択することが重要。

2.4 カラーモード

「画像を印刷したらディスプレイで見た色と違う色になってしまった」……そんな経験ないですか？ これは、ディスプレイとプリンタの色の表現方法が全く違うことに起因します。

ディスプレイは、赤 (レッド: Red)・緑 (グリーン: Green)・青 (ブルー: Blue) の3色の光で全ての色を表現します。この3色を光の三原色といい、レッドとグリーンを混ぜると黄 (イエロー: Yellow) に、グリーンとブルーを混ぜると緑青 (シアン: Cyan) に、ブルーとレッドを混ぜると赤紫 (マゼンタ: Magenta) に、3色全てを混ぜると白 (ホワイト: White) に、どの色も使わない、すなわち全く光がなければ黒 (ブラック: Black) になります。光の三原色は、各色の頭文字をとって RGB と呼ばれています。

一方プリンタは、シアン・マゼンタ・イエローの3色で全ての色を表現します。これを色の三原色 (色料の三原色 / 色材の三原色とも表現) といい、シアンとマゼンタを混ぜるとブルーに、マゼンタとイエローを混ぜるとレッドに、イエローとシアンを混ぜるとイエローに、3色全てを混ぜるとブラックに、どの色も使わなければ (一般的な紙の色である) ホワイトになります。色の三原色は、各色の頭文字をとって CMY と呼ばれています。ところが理論上は全ての色を表現できるこの CMY、実は3色を均等に混ぜても濁った暗い色になるだけで、綺麗なブラックになりません。そこで CMY に加えてブラックを使うことで、全ての色を表現しています。実際の一般的な印刷行程ではこの4色が用いられており、CMYK (K は black や Key plate の K) と呼ばれています。



カラーモードとは色の表現方式のことで、RGB や CMYK は代表的なカラーモードです。他にもよく使用されるものにグレースケールやインデックスカラーがあるので、以下にそれぞれの特徴を簡単にまとめてみます。

RGB

レッド・グリーン・ブルーの3色を使って色を表現する。1つの色を0~255の256段階で表現するので、RGBの3色で約1670万色 (これをフルカラーという) を表現できる。

CMYK

シアン・マゼンタ・イエロー・ブラックの4色を、それぞれ0~100%の値で指定し、色を表現する。

グレースケール

グレーの濃淡で色を表現する。最大256階調のグレーを表現できる。

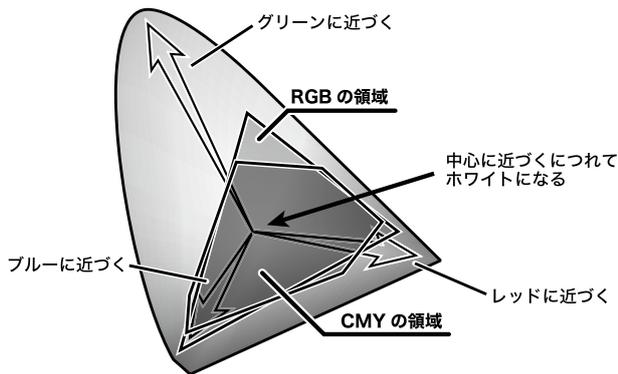
インデックスカラー

色と番号を結びつけて、番号で色を表現する。例えば「番号1は赤、番号2は紫、番号3は茶、……」としておき、「画素Aは2番、画素Bは1番、画素Cは3番、……」という具合に指定して表現する。最大で256色を表現できる。

では最初の問題に戻ってみましょう。普通PCで扱う画像は、ディスプレイに出力する前提で作られています。そのため、ディスプレイ出力に適したRGBカラーモードになっています。しかしプリンタはCMYKの4色で印刷するので、例えばR=192/G=128/B=64というように、RGBで表現された色情報なんて理解できません。そこでC=18%/M=55%/Y=80%/K=8%というように、RGBカラーモードからCMYKカラーモードに変換することで、プリンタが理解できる色情報にしてやるわけです。

しかしここで大きな問題があって、RGBで表現できる色の範囲と、CMYKで表現できる色の範囲は異なっており、CMYKはRGBよりもその範囲が狭いんです。つまりRGBだと表現できる色でも、CMYKでは表現できないこともあるってことです。そのためそのような色は、CMYKで表現できる最も近い色に置き換えられます。ところがRGB→CMYKの置き換えってのは、コンピュータがやるには意外と難しい処理で、色によっては人間の感覚で最も似ていると感じる色とは違った色に変換されてしまいます。このような過程を経ているので、ディスプレイとプリンタで色に違いが出ることもあるわけです。

一般的に、印刷した色はディスプレイの色に比べて彩度が下がって暗い感じの色になり、特に明るく鮮やかな色だと顕著に違いが出ます。



ディスプレイとプリンタで色の違いが出てしまうのを防ぐためには、どうすればよいのでしょうか？

・画像を0から作る場合

この場合は簡単で、プリンタで出力するつもりならCMYKカラーモードに、しないつもりならRGBカラーモードに設定してから画像を作れば問題ないです。

・RGB画像を印刷する場合

例えば「デジカメで撮った写真を印刷したい」とって場合がこれに当てはまります。この場合は、まずRGBカラーモードをCMYKカラーモードに変換できるソフトウェアを使い、カラーモードをCMYKにします。

変換してもそれほど色が変化しなかった場合は、そのまま印刷すれば問題ないです。しかし、彩度が下がって暗い感じになってしまった場合は、CMYKカラーモードに対応したソフトウェアを使って、なるべく元画像の色に近くなるように編集する作業が必要になります。簡単に言っていますが、これはなかなか難しい作業です。まずCMYKに変換したり編集したりできるソフトウェアがそんなに多くないですし、そもそも何もノウハウがない人がこの作業をやると思って、そうそうできるもんじゃありません。

こういった時には、写真屋や印刷会社に印刷を依頼するか、「そんなにこだわらなくてもいい写真だから、別に色が変わったっていいや」と割り切ってしまうでしょう。最近プリンタの性能も上がっているので、RGBカラーモードのまま印刷しても、それなりの再現度で印刷してくれます。でも所詮はそれなりなので、例えば青空や海みたいな写真だと、シヨンボりする印刷結果になってしまいます。

カラー画像で印刷をするつもりがないならRGBカラーにする。カラー画像で印刷をするつもりならCMYKカラーにする。白黒の画像ならグレースケールにする。

3 主な画像フォーマット

それじゃあ実際にどういう画像フォーマットがあつてどのように使われているかを見てみましょう。各フォー

マットには、扱えるデータの種類・圧縮方式・カラーモードを示しています。ちなみに、ある程度多くの人に使われていて、それなりに汎用的もしくは業界標準となっていて、かつ僕が知っている画像フォーマットしか載せていません。でもこれだけ知っていれば、困ることはあんまりないと思います。

JPEG (Joint Photographic Experts Group)

[ビットマップ][可逆圧縮, 不可逆圧縮][RGB, CMYK, グレースケール]

「恐らく現在世界で1番使われているのではないかな」と思ってしまうほど、(特にインターネットで)頻繁に見かけるフォーマットです。主な用途は、インターネット用のフルカラー画像やデジカメの画像なんですけど……もはや猫も杓子もJPEG画像、とりあえずインターネットで使うならJPEG画像にしとけて風潮です。こんな状況になってしまったのは、後述する理由でGIFが避けられたことと、PNGがそこまで普及しなかったことが挙げられると思います。

「JPEGは可逆圧縮にも不可逆圧縮にも対応してるようだし、こいつさえあれば別にいいんじゃないの?」と思うかもしれませんが……が、実はJPEGの可逆圧縮なんてあつてないようなもの。特許の関係でほとんど使われていません。よってJPEG画像といえば不可逆圧縮なんです。ここで問題になるのは、不可逆圧縮をすると画像が劣化してしまうということです。何でもかんでもJPEG画像にしていたら、写真のような画像はいいとして、CGのような画像では劣化が目立ってしまいます。

画像を圧縮する時は、まずGIFやPNGのような可逆圧縮のフォーマットにできないかと考慮してから、最終手段としてJPEGを使うようにしましょう。

GIF (Graphics Interchange Format)

[ビットマップ][可逆圧縮][インデックスカラー, グレースケール]

まだ回線速度がそれほど速くなかったインターネット初期において、Webブラウザ標準対応の画像フォーマットとしてJPEGと共に使われていたのがGIFです。写真のような画像は不可逆圧縮のJPEG、CGのような画像は可逆圧縮のGIFという住み分けでした。もちろん現在もインターネット用の画像フォーマットとして利用されていて、その大きな特徴は、何と言ってもアニメーションが可能なことと、透過色(透明色)が利用できることです。

アニメーションは、パラパラ漫画の要領でアニメーションを行うものです。透過色は、特定の1色を透明にして画像の背景を透過表示できるというもので、「画像をキャラに合わせて切り抜きたい」とって時なんかに使います。どちらもほとんどのWebブラウザが標準で対応している機能なので、「アニメーションを使いたい」「透過色を使いたい」とって時にはGIFを使いましょう。

ただしインデックスカラーかグレースケールか、つまり256色までしか扱えないので、色数が多い写真やグラデーションがあるような画像には不向きです。なので「フルカラー画像をアニメーションさせたい」って時はFlashを、「フルカラー画像を透過したい」って時はPNG（しかし後述するように問題がある場合もある）を使いましょう。もちろん、フルカラーは諦め、減色してGIFにするのも1つの手です。

さて、JPEGの項で触れたGIF離れについてですが、GIFはデータ圧縮にLZWって手法を使っています。そのLZWはアメリカのUNISYS社が特許を持っていたんですが、GIFが広まり出した頃からLZWの利用料を要求するようになりました。そのため、それまでGIFを利用していた人達は利用料の請求を恐れ、別のフリーなフォーマットを使うようになったというのが、GIFの避けられた原因です。そんなこんなで敬遠されたGIFですが、LZWの特許は2004年に失効したので、現在は安心して使うことができます。

PNG (Portable Network Graphics)

[ビットマップ][可逆圧縮][RGB, インデックスカラー, グレースケール]

PNGは、特許料の要求によって敬遠された、GIFの代替品となるべく作られたフォーマットです。GIFの代替なのでインターネットでの利用を想定しており、更にGIFの機能を色々と強化してあります。その主な特徴は

- ・可逆圧縮でフルカラーが可能
- ・一般的に同じ画像ならGIFより圧縮率が高い
- ・アルファチャンネルを持てる

といったことが挙げられます。アルファチャンネルとは透明度を表すもので、これによって、GIFでは透過色が1色でしかも透過するかしないかの2択しか選べなかったのに対して、PNGでは透明度を256段階で表せるようになりました。つまり半透明の表現が可能になり、徐々に透明にするといったこともできるようになりました。

しかし良いことばかりってわけではなく、GIFより劣っている点もあります。まずPNGはアニメーションを表現できません。PNGを更に拡張したフォーマットを使えばアニメーションも可能なんですが、Webブラウザが全く対応していないので、今のところほとんど使われていません。またPNGは比較的新しいフォーマットなので、PNGの機能全てに対応していないブラウザがあります。その中でも、現在最も多くの人に使われているInternet Explorer (IE) 6が、PNGのアルファチャンネルに対応していないというのが一番の問題です。ただこれに関しては、IE7を始め、最新のWebブラウザならPNGのアルファチャンネルに対応できているので、IE7が主流になる数年後にはこの問題は解決していることでしょう。

LZW圧縮の特許が失効した今、GIFの代替品として

の意義は失われましたが、それでもPNGは「インターネットで使える可逆圧縮でフルカラーを表示できる最も一般的なフォーマット」という重要な位置にあります。何でもかんでもJPEGにしようと思わず、CGのような画像なら、まずはPNGを使ってみましょう。画像は劣化しないし、色数次第ではJPEGよりもファイルサイズが抑えられるので、個人的にインターネットで使ってほしいフォーマットNo.1です。

BMP (Microsoft Windows Device Independent Bitmap)

[ビットマップ][無圧縮, 可逆圧縮][RGB, グレースケール]

Windows標準の画像フォーマットで、Windows同士でデータをやり取りするのなら、とりあえずBMPにしておけば古いPCでも問題なく見れます（古いMacintoshだと見れません）。

無圧縮と可逆圧縮があるんですが、可逆圧縮の方も圧縮率はそんなに高くないので、基本的にファイルサイズは結構大きくなります。そのため、インターネットで利用するにはあんまり適していません。また、CMYKカラーに対応していないので、印刷にも不向きです。自分のPC内でだけで使う画像や、画像を編集する際の中間フォーマット、古いWindowsとの互換性を保つために使うといった用途に向いています。

TIFF (Tagged Image File Format)

[ビットマップ][無圧縮, 可逆圧縮, 不可逆圧縮][RGB, CMYK, グレースケール]

"タグ"という識別子を使って、色んな形式のビットマップデータを表現できるフォーマットです。無圧縮だろうとLZW圧縮だろうとJPEG圧縮だろうともう何でもオケです。カラーモードもRGBにCMYKにグレースケールと何でもこい。あまりにも表現の幅が広すぎて、環境によっては完全な互換性がないってほどです。そのため、自分のPC以外の環境でも表示できるようにしたいなら、なるべく無圧縮で保存するのがベターです。

主なWebブラウザは標準に対応していないので、インターネットでの利用には向いていません。しかしTIFFはCMYKカラーに対応していて、かつ無圧縮もしくは可逆圧縮に対応しているので、印刷用途のビットマップ画像にとっても適しています。実際印刷業界では、ビットマップ画像についてはTIFFが標準的に用いられています。またBMPと同じく、(不可逆圧縮をしなければ)画像を編集する際の中間フォーマットにも向いています。他にも、(とりあえず無圧縮なら)古いWindows・Macintoshでも見れるので、どんな環境でも使える互換性の高い画像フォーマットとしても最適です。

WMF/EMF (Windows Metafile/Enhanced

Metafile)

[ビットマップ, ベクトル][RGB]

WMFはその名の通りWindowsが標準で扱えるフォーマットで、ビットマップデータとベクトルデータの両方を表現できます。しかしこいつはてんでダメなフォーマットです。なぜなら

- ・6万色程度までの色数しか扱えない
- ・一部のベクトルデータの描画に対応していない

という特徴があるからです。特に後者の欠点が致命的で、ベクトルデータをWMFで保存すると、線がガタついてしまうんです。そのためWMFなんてのは、正確な出力が求められない、どうでもいい画像にしか使うことができません。

一方EMFはWMFを拡張したもので、WMFでは扱えなかったフルカラーに対応し、ベクトルデータの精度も上がっています。しかし所詮ダメな奴を拡張してもダメな奴止まりで、結局ベクトルデータを正確に出力できません。そのためやっぱりどうでもいい画像に使うくらいしかできません。

何でこんなダメダメな奴らを紹介したかという、1つはビットマップとベクトルに対応しているものの、ベクトルデータを完全には再現できないという特徴を知ってほしかったから、そしてもう1つは、Microsoft Officeが標準で読み込める画像フォーマットの中で、ベクトルデータが扱えるのはWMFとEMFの2つだけだからです。ワードやエクセルにベクトル画像を貼りたい時には、結構便利かもしれません。それ以外の用途ならこんなフォーマットは用なしです。

ちなみに、それぞれの特徴を見れば分かるように、特に理由が無い限りはEMFの方を使うようにしましょう。

PS/EPS (PostScript/Encapsulated PostScript)

[ビットマップ, ベクトル][非圧縮][RGB, CMYK, グレースケール]

PSもEPSも、ビットマップ画像とベクトル画像を格納できて、更にRGB・CMYK・グレースケールに対応しているので、どんな画像を表現することができるフォーマットです。PSもEPSも使う分には同じようなものですが、両方使える場合なら普通はEPSの方を使います。

印刷用途を想定しているフォーマットなので、インターネットでは使われず、専ら印刷に利用されます。TIFFと同じく、印刷業界の標準フォーマットとなっていて、ベクトル画像を含んでいる場合はEPSが使われます。

PSやEPSは高品質な印刷を求める場合に使われるので、印刷物売り物とするような、画像をきれいに印刷する必要のある人以外は、特に触れることがないフォーマットだと思います。

SVG (Scalable Vector Graphics)

[ビットマップ, ベクトル][無圧縮, 可逆圧縮][RGB]
SVGは、XMLという言語で記述された汎用性の高い画像フォーマットで、ビットマップとベクトルの両方を表現できます。その最も大きな特徴は、Webブラウザでも閲覧可能だということでしょう。現在大半のブラウザが、標準で対応しているか、プラグインを導入することで、SVGを表示することができます。

今のところ、主要なWebブラウザが対応可能な、ベクトルデータを表現できる画像フォーマットはSVGだけです。インターネットでベクトル画像が使いたかったら、迷わずSVGを使いましょう。

PDF (Portable Document Format)

[ビットマップ, ベクトル][可逆圧縮, 不可逆圧縮][RGB, CMYK, グレースケール]

皆さん1度はインターネットでマニュアルや論文なんかの閲覧をした際に利用したことがあると思います。ブラウザクラッシュことPDFです。ファイルを開くに時間が無駄にかかるので嫌われがちですが、PDFはそれはもう素晴らしいフォーマットなんです。何が素晴らしいって、その可搬性の高さです。可搬性が高いってのはつまりどんな環境でも見られるってことで、

- ・レイアウトを保持できる
- ・フォントの埋め込みが可能

の2点がPDFの可搬性の高さを支えています。例えばワードのファイルだと、ワードがインストールされていない環境では見れないし、バージョンが違えばレイアウトが崩れてしまうこともあります。また、文書内で使っているフォントが閲覧側の環境にないと、代わりにのフォントで表示されてしまい、作成者の意図した表示・印刷ができません。一方PDFは、閲覧ソフトさえあれば、どんな環境でも同じように表示・印刷ができます。何故なら、PDFファイル自体にレイアウト情報やフォントデータが入っているからです。そしてPDFの閲覧ソフトは、Windows・Macintosh・Linuxなど、各種OS用のものが無償で配布されているので、大抵の環境なら困ることはないでしょう。

このような可搬性の高さに加え、PDFは文字・画像・図形など、ビットマップデータ・ベクトルデータを何でも表現でき、圧縮方式やカラーモードも大抵のものに対応しているので、

- ・不特定多数の人に渡したい
- ・複雑なデザインや標準にないフォントを使っている
- ・印刷用に使いたい

といったデータには、PDFを利用すると便利です。

4 おわりに

画像フォーマットとそれに関する知識を延々と書いてみましたが、「○○という概念が重要」「△△と□□を使い分けることが重要」「××するなら☆☆することが

重要」と重要なことばっかで、「お前は重要村の住人かよ！」と思うかもしれませんが。しかしこれでもエッセンスを厳選して書いたつもりです。要するに画像には、本当に重要なことがたくさんあるんです。ここで書いたことは、画像を上手に運用する上で恐らく避けては通れないでしょう。逆に言えば、画像を知り、画像フォーマットを知ること、状況に応じた画像の活用ができるようになるのです。

……とまあカッコいいことを言おうと思ったのですが、世の大半の人にとっては画像なんて見ればいいんですよ。それなりの画質があってファイルサイズが小さければ何でもいいんです。ベクトルだの解像度だの小難しいことなんてどうでもいいんです。多少画像がきれいになったって注意して見ないと全然気が付かないんです。というわけでJPEG最高！JPEGと水さえあれば生きていける！

この記事のまとめ

「画像なんて見ればいい」の精神でJPEGを使おう

| | データ | | 圧縮方式 | | | カラーモード | | | |
|------|--------|------|------|----|-----|--------|------|---------|--------|
| | ビットマップ | ベクトル | 無圧縮 | 可逆 | 不可逆 | RGB | CMYK | グレースケール | インデックス |
| JPEG | ○ | | | △ | ○ | ○ | ○ | ○ | |
| GIF | ○ | | | ○ | | | | ○ | ○ |
| PNG | ○ | | | ○ | | ○ | | ○ | ○ |
| BMP | ○ | | ○ | ○ | | | | | |
| TIFF | ○ | | ○ | ○ | ○ | | | | |
| WMF | ○ | △ | | | | △ | | | |
| EMF | ○ | △ | | | | ○ | | | |
| PS | ○ | ○ | ○ | | | ○ | ○ | ○ | |
| EPS | ○ | ○ | ○ | | | ○ | ○ | ○ | |
| SVG | ○ | ○ | ○ | ○ | | ○ | | | |
| PDF | ○ | ○ | | ○ | ○ | ○ | ○ | ○ | |