

コンピュータ内での画像処理について

60 回生 ばんだ

最近PCも普及して、みなさんもコンピュータを使う機会が増えてきたと思います。その中には、写真を取り込んで編集したり、年賀状の文面をドローソフトで作ったりと、画像を使う処理もたくさんあることでしょう。今日はその処理のことについて話したりしてみたりします。

コンピュータ内のデータの扱い

コンピュータの中では、データは全て2進数であらわされる、ということは聞かれたことがあるかもしれませんが。コンピュータ上でのデータサイズは、バイト (byte) という単位で表されます。1バイトは、8ビット (bit) で構成されています。1ビットというのは、そのデータが1か0か、ということなので、1バイトは、00000000~11111111まで、2の八乗の256通りの情報を表すことができます。(11110011、10010101、といった感じです) わかり易いように、00~FFまでの2桁の16進数で表すことが多いですね。

例を挙げると、「部誌の締め切りやべー」という文章は、テキストエディタ (メモ帳等の文章を扱うソフトです) で見ると、部誌の締め切りやべー、と見えますが、実際は、このようなデータになっています。

95 94 8E 8F 82 CC 92 F7 82 DF 90 D8 82 E8 82 E2 83 78 81 5B

また、データが大きな場合はK(キロ)やM(メガ)、G(ギガ)などをつけて表します。これは長さや重さでも使われるものと同じなのでわかりやすいと思います。「このパソコンのハードディスクは80GBだー」とか「オレのPCはメモリが512MBだぜっ」等と扱えるデータの量を表すときにも使われますね。

コンピュータ上での画像データ

コンピュータ上では、画像ももちろんデータの塊として扱われます。デジカメ等で取り込んだ画像も、ペインタ等で描いた画像も全部、同じようにファイルとして保存されます。

コンピュータ上での画像には、大きく分けて二つの形があります。1つがビットマップ画像、もう一つがベクタ画像です。

ビットマップ画像とは、画像を色のついた点(ドット)の羅列として表現したデータのことです。小さな点をものすごい数集めれば画像になりますからね。ラスタとも呼ばれることがあります。

ビットマップ画像は、写真などをあらわすのに良く使われます。1ドット単位で色を変えたりできるので、細かい描写に適しているからです。ただ、ドット単位でしかデータが入っていないため、極端に拡大するとドットが目立ちますし、逆に縮小するとドット当たりのデータが失われたりします。点を大量に集めているので、サイズもベクタ画像に比べるとかなり大きいことが多いです。

また、ビットマップ画像には解像度というものがあります。解像度は dpi(Dots Per Inch) で表されます。これはその画像を表示する際に、一インチ当たり何ドットが並んでいるかということで、パソコンのモニタは 72dpi です。(一インチに 72 ドットが並んでいる)印刷する時はそれより多く、300 ~ 400dpi ほどあればいいと言われてい

ます。

ベクタ画像とは、画像を座標や数式で定義したものです。

ベクタ画像は、縮小や拡大をしてもそれに合わせてコンピュータ側で組み替えられるので、情報が失われることはありません。また、ビットマップ画像に比べ編集が楽な部分もあります。

色の話

パソコン上の色には、主に 2 つのモードがあります。それぞれ、RGB、CMYK と言います。(他にもグレースケール等がありますが) RGB とは、光の三原色、R(Red)

G(Green) B(Blue) の三種類で色を表す方法です。それぞれの要素を最も強い光量で混ぜると白になる、加法混合という方法で混ぜられています。

3 つの要素を 0 ~ 255 までの量で表すのが一般的です。つまり、256 の 3 乗で 16777216 色が、理論上は表示可能です。

わかり易いように 16 進数で書くこともあります。

例えば黒なら R、G、B 全てが 0 なので 000000 と表されます。白は全てが 256 なので FFFFFFF という感じです。

他にも

- 赤:FF0000
- 黄色:FFFF00
- 抹茶:C5C56A
- 群青:4C6CB3

等と様々な色を表すことが出来ます。モニタは光で色を表しているの、モニタに出力する画像は基本的に RGB で作られます。

CMYK は、C(Cyan) M(Magenta) Y(Yellow) K(black) の四種類で色を表す方法です。RGB が光の三原色なのに対して、こちらは色の三原色 + 黒で色を表しています。純粋な C、M、Y を重ねると黒になるので、原色混合と言います。理論上は CMY のみで黒も表せるのですが、印刷時には黒を表すのが難しいので、黒を含めた四つの要素を使用します。ちなみに black が Kなのは Blue と間違えないようにしたからだとか。

CMYK は印刷に適しているので、印刷するとき等は基本的に CMYK を使います。また、CMYK は RGB よりも表示できる色数が少なかったりします。

プログラムの話

画像の加工

コンピュータ上では、データで画像が表されているので、加工がとても簡単です。例えばグレースケール化を考えて見ましょう。

ここでは、NTSC 係数による加重平均法と呼ばれる方法で考えて見ます。この方法は、R、G、B それぞれの値に重み付けをして 3 で割り、平均を取りグレースケール化する方法です。この方法以外にも、重みをつけない中間値法や単純平均法と言う方法もありますが、加重平均法のほうがより自然なグレースケールを作れると言われていています。

重み、と言うのは人間が色の違いによって感じる明るさの違いのことです。

例えば、FF0000 と 0000FF はコンピュータ上では同じ明るさですが、赤と青では人間は青のほうが暗く（重く）感じます。そこで青の明度を優先的に黒めに変換するようにするのです。

加重平均法では、グレースケール変換後の値 Y は

$$Y = (0.298912 * R + 0.586611 * G + 0.114478 * B)$$

と言う式で求めることができます。（R、G、B はそれぞれ前の色の R、G、B 成分）

画像の各ピクセルの色に対して全てこの処理を行うことにより、画像全体をグレースケール化することができます。

例えば、FF0000 と赤いピクセルは

$$0.298912 * 255 = 76.521472$$

の明度となりますね。

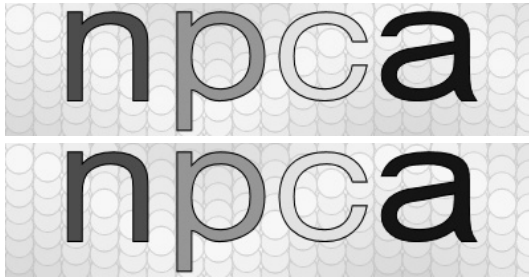
画像の圧縮

コンピュータ上で画像を扱う際、全ての画像をそのまま利用していたのではサイズが大きくなってしまいますので、そのサイズを小さくして使用しようと言うので、様々な圧縮法が考えられてきました。

ここで jpg の圧縮法について述べようかと思ったのですが、色々と事情があって（汗）代わりに各圧縮法のサンプルでも置いときます。



まずサンプル画像を bmp で用意してみました。サイズは 117 KB です。（ちなみに印刷じゃわかりにくいかもしれませんがカラーです）



jpg:62.1KB

png:29.1KB

とまあ圧縮したら画質はほぼ変わらず、かなりサイズが下がってますね。てなわけで、bmp のまま保存するのはあれなので、画像は圧縮して保存しましょう。ちなみに jpg は不可逆圧縮（元の画像からデータが一部失われているので完全には元に戻せない圧縮）なので、個人的には png のほうがお勧めですよ。

終わりに

てなわけでコンピュータ上での画像処理について書いてみました。なにぶん急いで書いたので結構かけなかったこととかもあったのですが、まあその辺は気合でどうにか・・・この記事を読んで、画像を扱うときに色々と考えたりしてみたりしてくれたりするとうれしかったです。