第 2 章

写真のデータを読み込もう

73 回生 2lu3

2.1　はじめに

C 言語を使って、BitMap という画像ファイルを読み込みます。

BitMap というのは、png や jpeg,gif などと同じようなものです。それぞれの違いは、どのようにしてデータを保存するかです。例えば、BitMap はドット (点) ずつに色のデータが集まっています。

> ■コラム: C 言語とは
>
> プログラミングをするための方法の一つです。パソコンの細かい動作までプログラミングをすることができますが、何かをするのにたくさん書かないといけないので敬遠されることもあります。詳しくは Wiki 検索をおねがいします。

2.2　どのようにデータが保存されているか

パソコンのデータの全ては、0 と 1 で表現されているのはご存知でしょうか？ ですが、001001010110 なんていうデータは人間には非常にわかりにくいです。

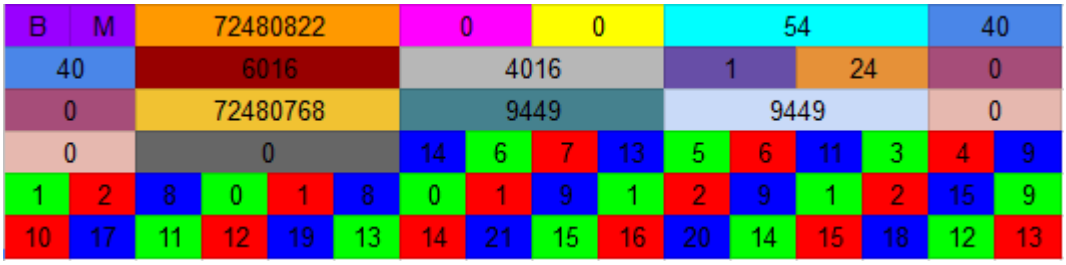
そこで、いくつかの数字を塊として考えるようにします。具体的には、8 個をひとかたまりにした 1byte(バイト) というものが使われています。8 個の桁それぞれで 0 か 1 を選ぶことができるので、2 の 8 乗個、つまり 256 通りものパターンを表すことができます。さて、下の図は 1byte ずつ BitMap の画像ファイルのデータを前から順番に読んでいったときの図です。ちなみに、2 進数での 1 桁をビット (bit) といいます。つまり、1byte は 8bit です。

この図の読み方は、下の図のとおりです。

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
49	50	51	52	53	54		56	57	58	59	60	61	62	63	64
65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96
97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112
113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128

さて、最初に [42] があり、次に [4D] があります。この [4D] は何でしょうか？ 正解は、**16 進数で表した数** です。今年は 2018 年ですが、この 2018 という数字は 10 進法で表されています。10 貯まるごとに、1 つ位を上げるから 10 進数というわけです。じゃあ、16 進数はというと、16 貯まるごとに 1 つ位を上げます。しかし、10 という数字を 1 桁で表すことはできません。そこで、0〜9 に加えて A〜F のアルファベットを使って 16 個を表します。

さて、16 進数のままだとわかりにくいので、文字 or10 進数に直した図を下に載せました。



	種類	バイト数	説明	備考
	ファイル部分	2	ファイルタイプ	常にBM
		4	ファイルサイズ	単位はバイト
		2	予約領域1	常に0
		2	予約領域2	常に
		4	画像データまでの距離	単位はバイト
	情報部分	4	情報部分のサイズ	単位はバイト
		4	画像の幅	単位はピクセル
		4	画像の高さ	単位はピクセル
		2	プレーン数	常に1
		2	1画素あたりの色数	ビット
		4	圧縮形式	普通は0
		4	画像データのサイズ	単位はバイト
		4	水平解像度	
		4	垂直解像度	
		4	パレットの色数	
		4	重要なパレットのインデックス	
	画像データ部分	1	青の輝度	
		1	緑の輝度	
		1	赤の輝度	

2 つの画像の色は対応しています。これから、それぞれについて解説をしていきます。

ファイルタイプ

BitMap なので「BM」というわけです。

ファイルサイズ

言葉の通り、ファイルの大きさを byte で表します。この BitMap 画像の場合、72,480,822byte、つまり約 72Mbyte になります。

予約領域 1・2

おまじないのようなものです。

画像データまでの距離

ファイル部分と情報部分のバイト数を全て足してみて下さい。54 になります。要するに、一番最初から画像データ部分まで何バイトあるかという話です。

情報部分のサイズ

情報部分のバイト数をすべて足すと、40 になりますね。

画像の幅

ピクセルというのはご存知でしょうか？ 簡単にいえば、1 ピクセルに 1 つ色があって、ピクセルが集まることによって画像になります。

近くからみた蟻の列は途切れているように見えても、遠くから見た蟻の列は一本の線に見えます。つまり、近くから見たらてんでばらばらな色があるようにしか見えなくても、遠くからみたら一枚の写真に見えます。

さて、話が脱線しましたが、画像の横方向にピクセルがいくつあるのかがここに入っています。

画像の高さ

縦方向にピクセルがいくつあるのかがここに入っています。

プレーン数

これは、よくわかりませんが
現在では使われなくなった概念です by Wiki(意訳)
とのことなので、気にする必要はありません。ちなみに、常に 1 です。

1 画素あたりの色数

正確には、色ビット数と呼びます。24 のときは、1677 万色が使えるそうです。正直、全部使うことはないでしょう。

圧縮形式

どのようにして圧縮をするかです。

画像データのサイズ

画像データ部分のデータの大きさです。画像データのサイズと画像データまでの距離を足すと、ファイルサイズになることをお確かめ下さい。

水平解像度・垂直解像度

> 水平解像度（すいへいかいぞうど）とは、アナログ放送時代のテレビ・ビデオなどの映像機器の、画質の指標のひとつである。　参照元:Wiki

とのことです。

パレットの色数

1 画素あたりの色数が 1,4,8 のときに使うらしいです。上で書いたとおり、正確には色ビット数と呼びます。それぞれ、1bit：2 種類 4bit：16 種類 8bit：256 種類色を登録することができます。

重要なパレットのインデックス

0 の場合もあるらしいです。使ったことはありません。

青・緑・赤の輝度

色の三原色はご存知でしょうか？ 赤 (Red)・緑 (Green)・青 (Blue) で、RGB とまとめて呼ばれることもあります。そして、RGB のそれぞれの色の濃さによって色を表現することができます。ここでは、1 ピクセルごとに色を決めています。

2.3　実際の読み方

C 言語でやります。
やることは、順番に fread() で読み込んでいくだけです。下にサンプルコードをおいておきます。

```
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <Windows.h>
#define true 1
#define false 0
#define HEADER_SIZE 54 //heder size 54 = 14 + 40
#define MAX_WIDTH 3000 //upper limit of width(pixel)
#define MAX_HEIGHT 3000 //upper limit of height(pixel)
#define COLOR_CATEGORY_NUM 1000
#define PLUS_MINUS(a, b, c) ((a) + (c) >= b && (a) - (c) <= b)
#define SIZE 10

// r,g,b values of one pixel
typedef struct {
  unsigned char r;
  unsigned char g;
  unsigned char b;
  int color_type;// どの色か
  int color_num;// color_categoryの場合、どれだけのピクセルがその色に入っているのか
  int sum;// r + g + b;
} color;

typedef struct {
  long height;
  long width;
  color data[MAX_HEIGHT][MAX_WIDTH];
  int color_num;// 色の数
  color color_category[COLOR_CATEGORY_NUM];
} img;

int ReadBmp(char *file_name, img *imgp) {
  unsigned char bmp_header_buf[HEADER_SIZE]; // this store header data of bmp
  char bmp_type[2];
  long bmp_height;
  long bmp_width;
  unsigned short bmp_color;
  long real_width;

  unsigned char *bmp_data;
  FILE *bmp_file;

  bmp_file = fopen(file_name, "rb");
```

```
  if(bmp_file == NULL) {
    printf("\n\n\n\nFailed to open bmp file\n");
    return 0;
  }

  // ヘッダー(両方)を取得
  fread(bmp_header_buf, sizeof(unsigned char), HEADER_SIZE, bmp_file);

  // ファイルの最初がBMかどうか
  memcpy(&bmp_type, bmp_header_buf, sizeof(bmp_type));
  if(strncmp(bmp_type, "BM", 2) != 0) {
    printf("Error: %s is not a bmp file.\n", file_name);
    return 0;
  }

  memcpy(&imgp->width, bmp_header_buf + 18, sizeof(bmp_width));
  memcpy(&imgp->height, bmp_header_buf + 22, sizeof(bmp_height));
  memcpy(&bmp_color, bmp_header_buf + 28, sizeof(bmp_color));
  if(bmp_color != 24) {
    printf("Error: bmp_color = %d is not implemented in this program.\n", bmp_color);
    return 0;
  }
  if(imgp->width > MAX_WIDTH) {
    printf("Error: bmp_width = %ld > %d = MAX_WIDTH\n", imgp->width, MAX_WIDTH);
    return 0;
  }
  if(imgp->height > MAX_HEIGHT) {
    printf("Error: bmp_height = %ld > %d = MAX_HEIGHT\n", imgp->height, MAX_HEIGHT);
    return 0;
  }
  if(imgp->height < 0) {
    printf("Error hight is under 0\n");
    return 0;
  }

  printf("Finished checking file format\n");

  real_width = imgp->width * 3 + imgp->width % 4;//calculate real width to fit 4 byte border


  if((bmp_data = (unsigned char *)calloc(real_width, sizeof(unsigned char))) == NULL) {
    printf("Eroor: memory allocation failed for bmp_data\n");
    return 0;
  }

  printf("Reading data...\n");
  for(int i = 0; i < imgp->height; i++) {
    fread(bmp_data, 1, real_width, bmp_file);
    for(int j = 0; j < imgp->width; j++) {
      imgp->data[imgp->height-i-1][j].b = bmp_data[j*3];
      imgp->data[imgp->height-i-1][j].g = bmp_data[j*3+1];
      imgp->data[imgp->height-i-1][j].r = bmp_data[j*3+2];
      imgp->data[imgp->height-i-1][j].sum = bmp_data[j*3] + bmp_data[j*3+1] + bmp_data[j*3+2];
    }
  }
  printf("Finished Reading data\n");

  printf("Please wait a few seconds...\n");

  free(bmp_data);
  fclose(bmp_file);
  return 1;
}



int main() {
  FILE *color_data_file;
  FILE *out_put_data_file;
  img *receiver_data;
  receiver_data = (img *)malloc(sizeof(img));

  char file_name[100];
  // Get file name
  printf("Drag and drop a bmp picture which you want to convert\n");
  printf("And push enter key if I don't say start opening\n");
  // If the name of picture have \n or ", it will be failed.
  scanf("%s", file_name);

  if(!ReadBmp(file_name, receiver_data)) {
    //unsuccess
    return -1;
  }
  return 0;
}
// ご苦労さま
```