

生産専攻 2 年前期 画像情報処理工学

第3回目の目標 : ppm 形式カラー画像データを作成できる。プログラムやデータファイル名 pc

1 前回までの解説

以下のホームページに前回の資料を掲載している。

URL: <http://www.ichinoseki.ac.jp/satok/SATOK/ex/index.html>

前回の授業のキーワード

領域塗りつぶしによるモノクロ画像の生成、数学関数によるモノクロ画像生成

今回は、インターレース形式のカラー画像作成を行う。

2 カラー画像(ppm 形式)を作成するサンプルプログラム pc1.c

このプログラムをコピーして一部改良し、改良後は課題番号に従って pc2.c, pc3.c としてファイル保存。画像ファイルも、この課題のものであることがわかるように pc1.ppm というように、プログラムに対応した名称とする。これらは後で、USB メモリ等でファイル提出を依頼することがある。

カラー・インターレース画像の ppm 保存形式。

(640x400 サイズで 1 画素8ビットの場合)

P6 ←— 最初に P6 の文字キャラクターがありカラー画像の raw 画像であることを示す

INCT ←— コメント(何文字でも可)

640 400 ←— 画像サイズの文字 横(ピクセル)が 640、縦(ライン)400 の画像データ

255 ←— 整数最大値が 255 であり、8 ビット画像データである

(この後に 8 ビットカラーのインターレース画像データが 640x400x3 バイト続く。

最初が 255 0 0 なら、赤を示す。インターレースの並びは RGB の順となる)

ppm 画像(pgm, ppm)の保存形式の解説はこちら

<http://www.mm2d.net/c/c-08.shtml>

<http://www.aso.ecei.tohoku.ac.jp/~kato/gzipped-pnm/gzipped-netpbm-ja.html>

<http://www.media.eng.hokudai.ac.jp/~chicchi/H14/exam1/chapter05.html>

以下のプログラム pc1.c は、図1で示す4つの領域それぞれに、

白(画素値 R=255,G=255,B=255)

赤(画素値 R=255,G=0B=0)

緑(画素値 R=0,G=255,B=0)

青(画素値 R=0,G=0,B=255)

で塗りつぶす目的のものである。

なお、図の輪郭線は、白地図形がわかるようにしたもので、実際には存在しない。

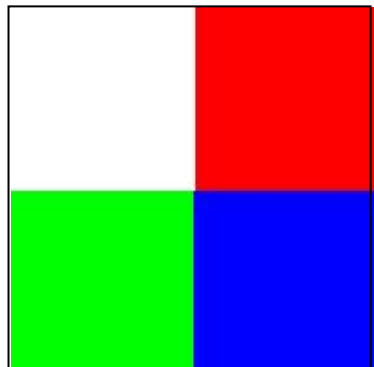


図1 カラー画像生成例 pc1.ppm

```

/* -----
Image Processing pc1.c
Color Image Pattern Generation
Quad area paint image white/red/green/blue
Ichinoseki National College Advanced Course
----- */

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

main(argc,argv)
int argc;
char *argv[];
{
    static int i,j,k,m;
    static int pixel,line;
    FILE *fp1;

    /* Command parameter check */
    if (argc!=4){
        printf("Usage : command pixel line output_file¥n");
        return 1;
    }

    /* Image size parameter load */
    sscanf(argv[1],"%d",&pixel);
    sscanf(argv[2],"%d",&line);

    /* Output image file name read */
    if ((fp1=fopen(argv[3],"w"))==NULL){
        printf("Can't open output_file¥n");
        return 1;
    }

    /* ppm header output */
    fprintf(fp1,"P6¥n"); /* インターレース形式のカラー画像 */
    fprintf(fp1,"# INCT¥n");
    fprintf(fp1,"%s %s¥n",argv[1],argv[2]);
    fprintf(fp1,"255¥n"); /* R,G,B は 8 ビットデータ */

    /* Color image generate */
    for (j=0; j<line; j++)
    {
        for (i=0; i<pixel; i++)
        {
            if (i<=pixel/2 && j<=line/2){
                putc(255,fp1); /* R データ */
                putc(255,fp1); /* G データ */
            }
        }
    }
}

```

```

        putc(255,fp1); /* B データ 以上の結果、この領域の色は白となる*/
    }
    else if (i>pixel/2 && j<=line/2){
        putc(255,fp1);
        putc(0,fp1);
        putc(0,fp1); /* 赤で塗りつぶす */
    }
    else if (i<pixel/2 && j>line/2){
        putc(0,fp1);
        putc(255,fp1);
        putc(0,fp1); /* 緑で塗りつぶす */
    }
    else{
        putc(0,fp1);
        putc(0,fp1);
        putc(255,fp1); /* 青で塗りつぶす */
    }
}
}
fclose(fp1);
return 0;
}

```

3 画像データを作成し、ファイルを観察してみる

1) サンプルプログラムによる画像ファイルの生成

```
pc1 200 200 pc1.ppm
```

8ビット、200x200 サイズのカラー画像 pc1.ppm を作成する。生画像データのサイズは 120000 バイト。

2) データサイズはいくらか？

前回と同様に `ls -l` コマンドで `pa1.pgm` ファイルのサイズを確認してみよう。何バイトであったか？

40000 を超えたバイト数が、上のリストの `/* pnm(pgm) header code output */` 部分で追加した画像形式を示す目的のヘッダー情報の容量である。

3) 画像データの中身の表示 (od コマンド)

```
od -x pc1.ppm | more
```

で 16 進数の画像データファイルを表示

```
od -c pc1.ppm | more
```

で文字による画像データファイルを表示

3) GIMP による表示観察

画像ファイルを「開く」で `pc1.ppm` ファイルを探し、表示してみよう

授業では ppm 形式のまま扱うが、必要に応じて jpeg 画像等に形式変更して保存できる。

4) 画像ファイルサイズを変更し、いろいろな画像を作成してみる

```
pc1 100 100 pc1.ppm
```

100x100 サイズの画像になる

```
pc1 640 400 pc1.ppm
```

640x400 サイズの画像になる

4 本日のプログラムの改良課題(必須)

残す時間を、上のサンプルプログラムを改良して、次のプログラム及び画像ファイルを作りなさい。

(1) 図2のように8色の色を塗りつぶすプログラム pc2.c を作成し、200x200 サイズの画像 pc2.ppm を生成しなさい。なお塗る色は左から R,G,B の順番で、次のようにする。

	R	G	B
紫	255	0	255
青	0	0	255
薄青	0	128	255
緑	0	255	0
黄緑	128	255	0
黄	255	255	0
橙	255	128	0
赤	255	0	0

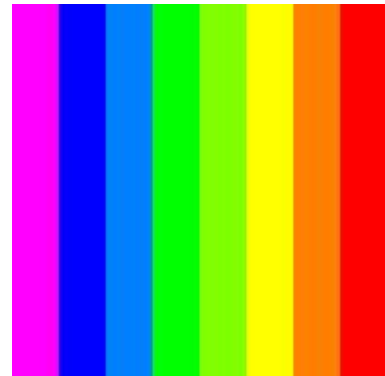


図2 8領域の塗りつぶし

【参考】

座標値の条件により色塗りする場合、if 文で切り替える方法と、以下のように switch, case で切り替える方法もある。以下の例は、4領域それぞれに色塗りする場合の切り替え例を示す。

```
switch(i*4/pixel){
case 0: /* 4 領域にわけた場合の左端を意味する */
putc(90,fp1); putc(62,fp1); putc(255,fp1); break;
case 1:
putc(45,fp1); putc(255,fp1); putc(0,fp1); break;
case 2:
putc(128,fp1); putc(128,fp1); putc(92,fp1); break;
case 3:
putc(255,fp1); putc(255,fp1); putc(255,fp1); break;
}
```

(2) 連続階調の画像生成

図3は、R,G,B の値を座標値に対し連続的に変化させた場合の画像である。この画像を生成するプログラム pc3.c と 200x200 サイズの画像 pc3.ppm を作成しなさい。

各色の値の式

ピクセルの値を $x(0 \leq x < \text{pixel})$, ラインの値を $y(0 \leq y < \text{line})$ とし、 $x=0, y=0$ を左上隅の位置とする。

$R = ax$ とし、 $x=\text{pixel}$ で $R=255$ とする。

$G = bx+c$ とし、 $x=0$ で $G=255$, $x=\text{pixel}$ で $G=0$ とする。

$B = dy$ とし、 $y=\text{line}$ で $B=255$ とする。

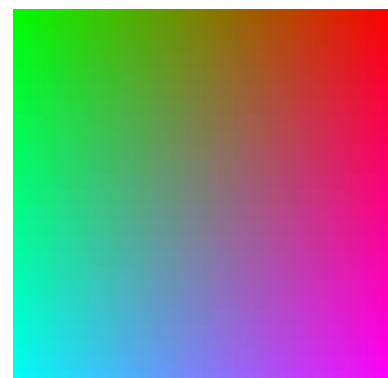


図3 連続階調によるカラー画像

5 プログラムの応用課題

これは自学自習課題である。

(1) 罫線を引く。

図4は、図3の画像に、20ピクセルごとに白い罫線を引いた画像である。

こうなるようなプログラム pc4.c を作成し、200x200 サイズの画像データ pc4.ppm を作成しなさい。

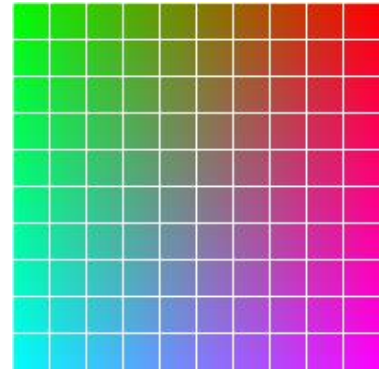


図4 カラー画像に罫線を追加

(2) カラー画像の切り抜き

図4は、図3の画像(サイズは200x200)に対し、半径100の円で切り抜いたものである。

円の方程式が、

$$x^2 + y^2 = r^2$$

であることに注意して、図5の画像を生成するプログラム pc5.c と画像データ pc5.ppm を作成しなさい。

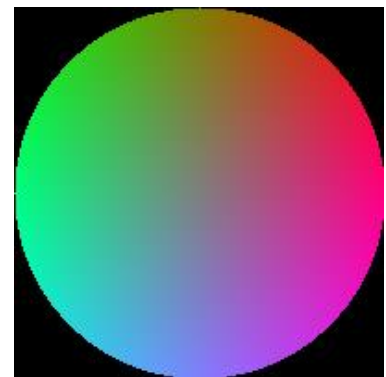


図5 カラー画像の円形切り抜き

この応用課題は、期末試験の問題のひとつと考えている。
グループで開発してもよいが、どんな原理で画像作成しているか、各自、なぜそうなるか、理由がわかっていることが大事である。