

1 グループについて

グループ C
s153038 木戸 喜隆
s153071 長迫 強士

2 プログラムファイルの実行方法

プログラムの実行については Makefile を作成した。以下の操作を行い実行する

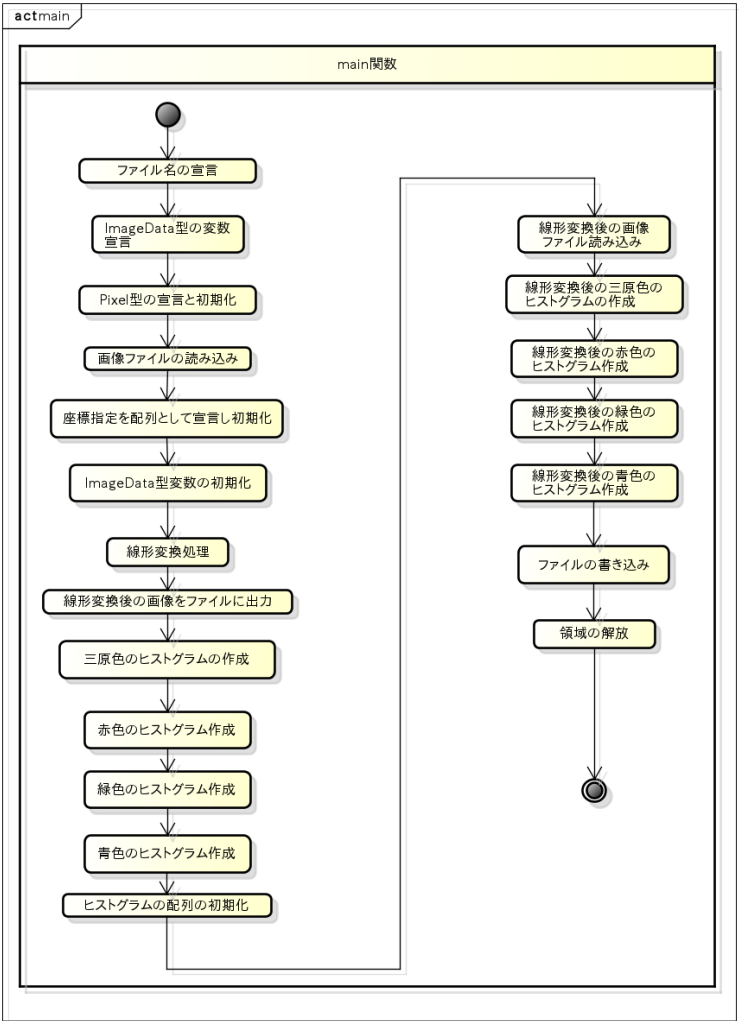
```
s153071@av0025:~/classes/multimedia/11/Multimedia_Exercise/group_work$ make
gcc -o 2017multiC 2017multiC.c bmp.c image.c
s153071@av0025:~/classes/multimedia/11/Multimedia_Exercise/group_work$ ./2017multiC
read[rgb.bmp]
read[out.bmp]
s153071@av0025:~/classes/multimedia/11/Multimedia_Exercise/group_work$
```

2.1 プログラムリスト

ソースコード 1 Makefile

```
1 2017multiC: 2017multiC.c bmp.c image.c
2 gcc -o 2017multiC 2017multiC.c bmp.c image.c
```

3 プログラムの構造説明



3.1 ヘッダ部 (1 から 13 行目まで)

ヘッダ部ではヘッダファイルの呼び出しとヒストグラムの画像の大きさなどのマクロとプロトタイプ宣言を記述した。また、ヒストグラムで用いる変数を二次元配列グローバル変数にすることでファイル内であればどこでも使用できるようにした。関数については ImageData 型の変数については省略している。

3.2 main 関数

15 から 48 行目まで

17 から 31 行目までは変数の宣言を行い、33 行目から 48 行目までは画像ファイルの読み込みや初期化などを行っている。また、ImageData 型変数の初期化では画像を連結するために以下のように初期化している。

- outimg :元の画像と線形変換後の画像の比較を行うため横幅を 2 倍にする
- outlinear :線形変換後のデータを保持
- outrgb :ヒストグラムの画像データの保持
- outrgb2 :元の画像と線形変換後の画像の赤、緑、青のヒストグラムの比較を行うため横幅 2 倍縦幅 3 倍にする
- outcolor :元の画像と線形変換後の画像の三色合成のヒストグラムの比較を行うため横幅を 2 倍にする

50 から 93 行目まで

51 行目から 53 行目までは linear 関数を呼び出し線形変換後の画像を作成してからファイルに出力している。59 行目では save 関数を呼び出し最初に読み込んだ画像データを連結する画像データに保存している。

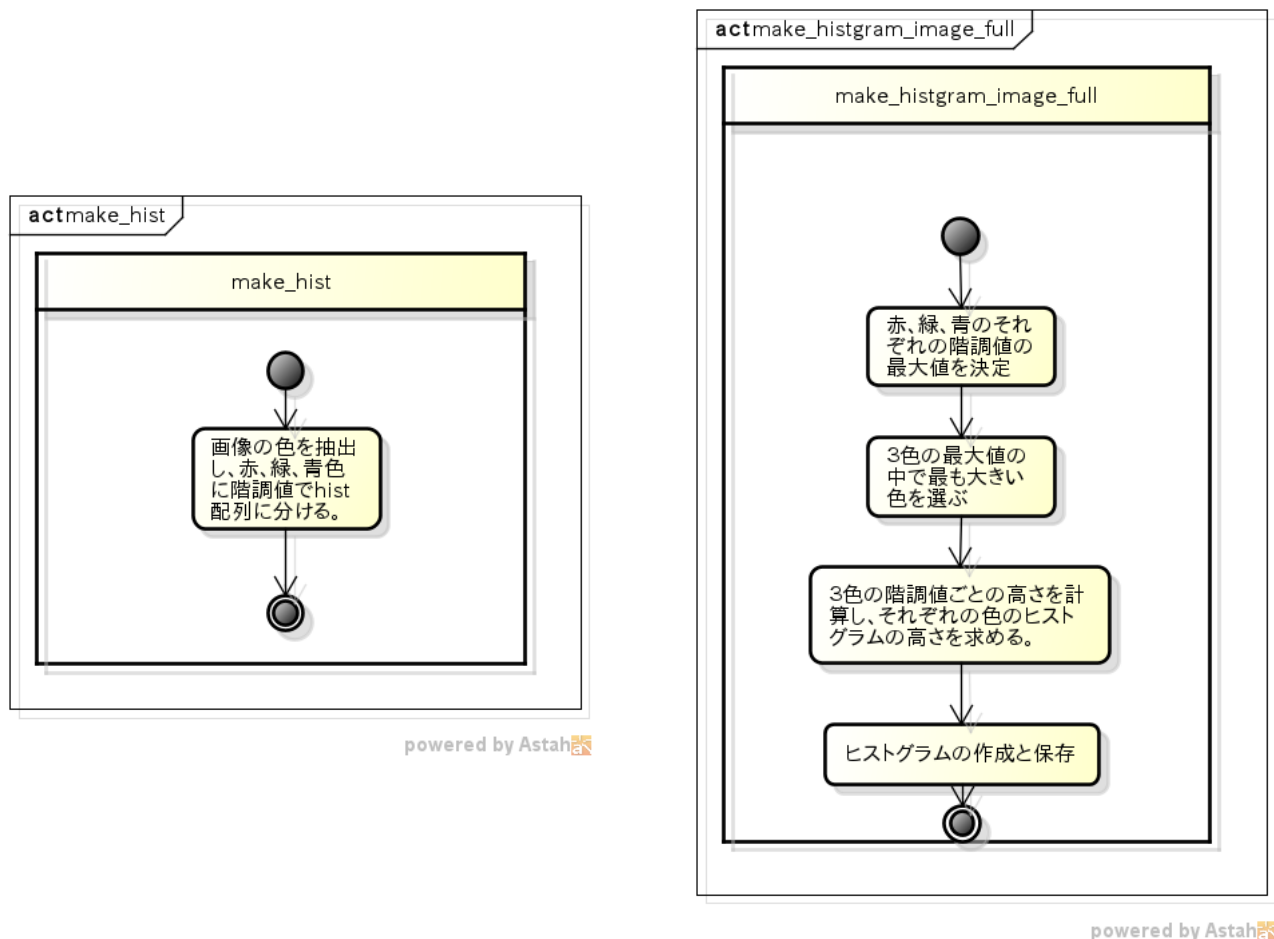
99 から 148 行目まで

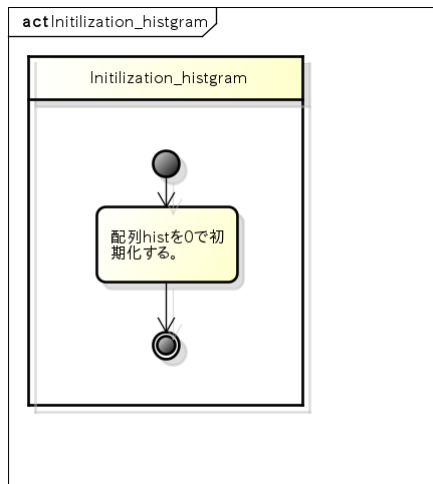
99 行目でヒストグラムの初期化をする。103 行目で線形変換後のファイルを読み込み、線形変換後の画像とヒストグラムを保存している。

150 から 161 行目まで

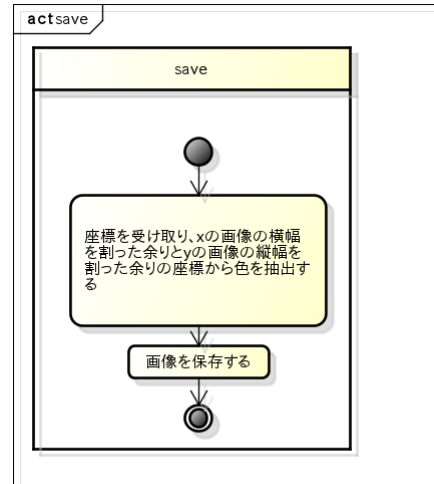
ファイルの書き込みと ImageData 型と Pixel 型の領域の解放を行っている。

3.3 関数について

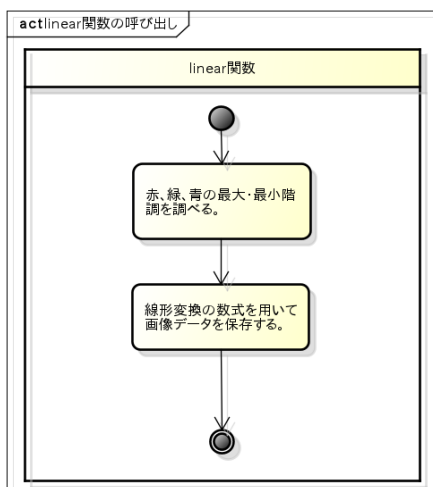




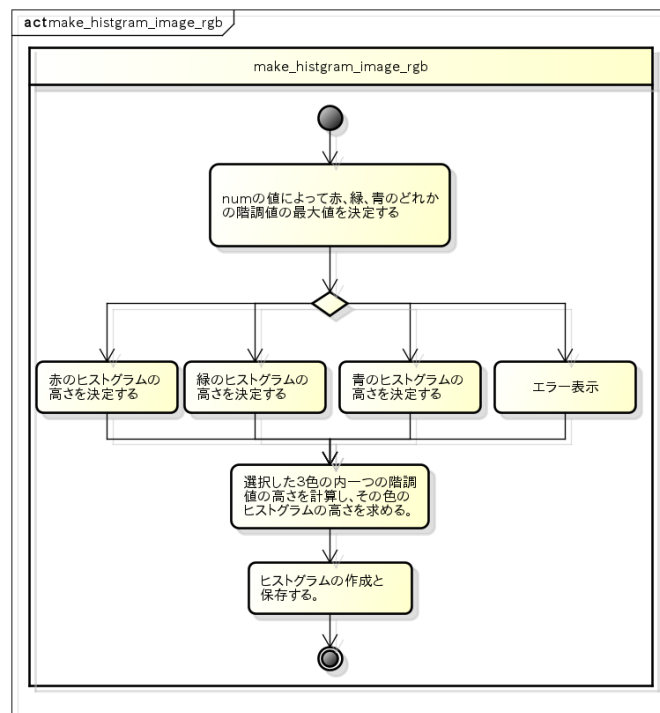
powered by Astah



powered by Astah



powered by Astah



powered by Astah

make_hist 関数

画像の色を getPixel 関数で抽出し、赤、緑、青のそれぞれの階調値に対応した hist 配列にインクリメントする。

make_histogram_image_full 関数

三原色それぞれの階調値の最大値の決定をして、その中で最も多い階調値を決定する。そして、三原色それぞれのヒストグラムを一つの画像に作成する。

make_histogram_image_rgb 関数

num の値により赤、緑、青のどれか一つの階調値の最大値を決定し、switch 文で選んだ色の最大のヒストグラムの高さを決定する。選んだ色のヒストグラムを作成し、画像データの保存を行う。

Initialization_histogram 関数

hist 配列を 0 で初期化する。

save 関数

データを保存する関数で width 配列で横幅、height 配列で縦幅を指定し、そこに画像データを保存する。

linear 関数

value は現在の階調値、max は最大階調値、min は最小階調値、new value は線形変換後の階調値として

$$newvalue = \frac{value - min}{max - min} \times 255$$

の式を用いて、三原色それぞれ行った。

4 グループとしてのプログラムの工夫

4.1 関数に分けてプログラムをわかりやすくした

make_hist 関数や make_histogram_image_full(または rgb) 関数に分けることでヒストグラムの作成する方法を細分化し、ソースコードが読みやすくするようにできる限り分かりやすくした。full と rgb を分けたのは感覚的に理解してもらうためである。また save 関数、linear 関数、Initialization_histogram 関数なども同様である。

4.2 画像を連結することで画像の変化を比較しやすくした。

元の画像と線形変換後の画像、その 3 色合成したヒストグラム、赤、緑、青の 3 つに分けたヒストグラムをそれぞれ 3 つの画像に連結してデータを保存することで比較ができるようになった。左が変換前の画像で右が変換後の画像である。

4.3 連結するとき配列を用いて座標の指定を行い好きな場所に保存することができるようにした

同じサイズの画像同士であれば width 配列と height 配列による座標の指定をすることで画像のどこにでもデータを保存することができる。また感覚的にわかりやすくするため数の小さい方を画像データの座標の最小値、大きい方を画像データの座標の最大値としている。今回使用しなかったが画像の拡大縮小をすることができ、num の値により画像サイズを変えることができる。ただし、画像は荒くなる。