pgmcat プログラム内部設計書

B0246810 徳久雅人(2004.5.24)

1. はじめに

pgmcat 外部設計によると, 外部機能には次の8つがある.

(ｱ) コマンドラインオプションを解析する機能
(ｲ) ﾘｽﾄﾌｧｲﾙからﾌｧｲﾙ名を抽出する機能
(ｳ) ファイル名の条件を検査する機能
(ｴ) 画像ファイルを読み込む機能
(ｵ) 画像ファイルの明るさを計算する機能･集計する機能
(ｶ) 画像サイズの高さを計算する機能･集計する機能
(ｷ) 画像の高さを指定して, 縮小する機能
(ｸ) 画像を横並び連結で出力する機能

これらの機能の内部設計を行う.

2. 内部設計

2.1. プログラム構造設計

pgmcat のモジュール構成を図1に示す.

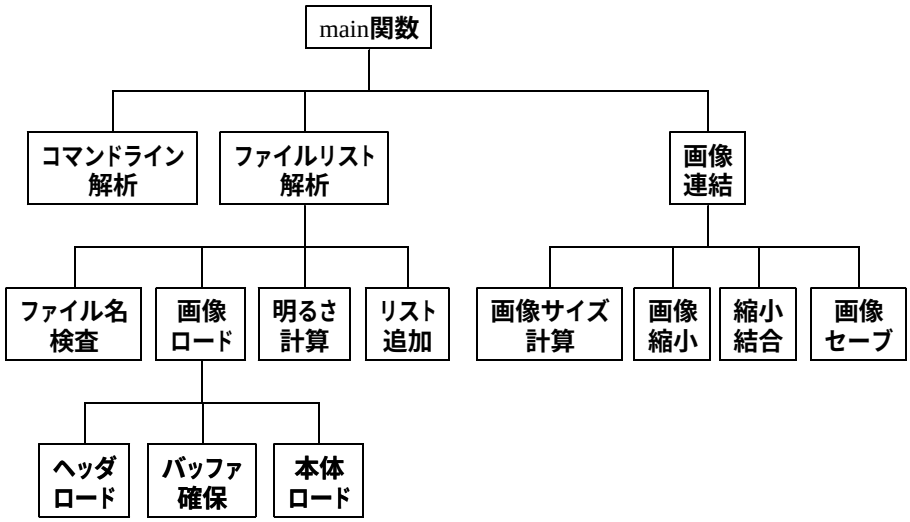

図1：pgmcat のモジュール構成図

2.2. プログラム内設計

図1に示したとおり, ﾒｲﾝ関数は, ｺﾏﾝﾄﾞﾗｲﾝ解析, ﾌｧｲﾙﾘｽﾄ解析, および, 画像結合の3つに分割する. 概略を説明する. ｺﾏﾝﾄﾞﾗｲﾝ解析の結果, 抽出するﾌｧｲﾙ名の条件, および, 明るさの条件が得られるので, ﾌｧｲﾙﾘｽﾄ解析において, 画像を選択的に蓄積する. 画像連結では, 蓄積した画像の最小の高さのサイズを計算し, 縮小して結合させて出力する. 出力先は, ｺﾏﾝﾄﾞ解析の結果により, 標準出力であるかﾌｧｲﾙ出力であるかが定まる.

2.2.1. **コマンドライン解析**

●コマンドライン解析

【関数名】 void analyse_command(int argc, char **argv, char *cond_name, int *cond_light, FILE *output);

【入力条件】 argc 引数の個数
argv 引数の文字列

【出力条件】 cond_name ファイル名の制約文字列. NULL ならば制約なし.
cond_light 輝度の制約値. 負の値ならば制約なし.
output 出力先のファイルポインタ.
STDOUT または オープンしたファイル.

【アルゴリズム】 出力値のデフォルト値を代入する. argc の数に応じて文字列比較を行う. 引数の出現順序が固定であるため, 条件文の列挙により判定する.

2.2.2. **ファイルリスト解析**

●ファイルリスト解析

【関数名】 void analyse_filelist(char *cond_name, int cond_light, Node *head);

【入力条件】 cond_name ファイル名の制約文字列. NULL ならば制約なし.
cond_light 輝度の制約値. 負の値ならば制約なし.
head リスト構造の先頭ノードを示すポインタ

【出力条件】 head 連結の対象となる画像情報のリスト.

【アルゴリズム】 (1) 標準入力のある間, 以下を実行
(2) 標準入力よりファイルリストの内容(ファイル名)が1行ずつ入力される. 各行(各ファイル)について以下を実行する.
(3) ファイル名の先頭文字列が条件の文字列と一致しないならば, (1)へ.
(4) 画像をロードする.
(5) 明るさを計算する.
(6) 輝度値の条件を満たさないならば, (1)へ.
(7) 画像サイズ, 画像データを格納するノードを作る.
(8) リストに追加. (1)へ.

●ファイル名検査

【関数名】 int check_filename(char *filename, char *cond_name);

【入力条件】 filename 検査されるファイル名
cond_name ファイル名の制約文字列. NULL ならば制約なし.

【出力条件】 int 返り値 条件を満たすならば 1 を返し, 満たさなければ 0 を返す.

【アルゴリズム】 ファイル名と制約文字列を文字列の先頭から比較する.

●画像ロード

【関数名】 void load_image(char *fname, unsigned char *buff, int *sizex, int *sizey)

【入力条件】 fname ロードする画像のファイル名

【出力条件】 buff この関数内でメモリを確保し, 画像のピクセル値を1次元配列に格納している.

size, sizey ロードした画像のサイズ.

【アルゴリズム】 (1) ファイルを開く.

(2) pgm ファイルのヘッダをロード. sizex と sizey を得る.

(3) バッファを確保する.

(4) pgm ファイルの本体をロード.

●明るさ計算

【関数名】 int calc_ave_of_light(unsigned char *buff, int sizex, int sizey)

【入力条件】 buff 画像データ

sizex, sizey 画像サイズ

【出力条件】 int 返り値 平均の輝度値

【アルゴリズム】 画像データの輝度値の平均値を求める.

2.2.3. **画像結合**

●画像連結

【関数名】 void cat_images(Node *head, FILE *output)

【入力条件】 head 結合すべき画像の線形リスト

output 出力先ファイルデスクリプタ. 書き込みオープンされたもの, または, stdout .

【出力条件】 (output に指定したところに結合した画像を出力する)

【アルゴリズム】 (1) リスト (head) 内の画像を1つずつ調べ, 画像の高さの最小値を求める.

(2) リスト内の各画像に対して, 画像の高さを指定して, 縮小する.

(3) リスト内の各画像を横に並べて結合した画像を作る.

(4) 画像を output を通じて出力する.

●画像サイズ計算

【関数名】 int measure_min_of_sizey(Node *head);

【入力条件】 head 画像の線形リスト

【出力条件】 int 返り値 画像の高さの最小値

●画像縮小（縦方向を基準にした全体画像の縮小）

【関数名】 void minimize_sizey(Node *head, int minsize)

【入力条件】 head 画像の線形ﾘｽﾄ

minsize 画像の縦サイズの目標値

【出力条件】 （画像ﾘｽﾄ内の各画像において，縦･横のサイズ，画像データが書き換わる）

【アルゴリズム】 (1) ﾘｽﾄ(head)内の各画像について以下を行う．

(2) 画像の縦サイズ ／目標値の比を求める．

(3) 縮小後の画像のﾋﾟｸｾﾙの位置を (x,y) として，各輝度値について以下を行う．

(4) (2)で求めた比に基づき，位置(x,y)が対応する元画像の位置を求める．

(5) 位置(x,y)に (4)で求めた位置の輝度値を代入する．

(6) (2)で求めた比に基づき，横サイズを求める．

(7) ﾉｰﾄﾞの画像サイズを書き換える．

●画像結合

【関数名】 void cat_images(Node *head, int *sizex, int *sizey, unsigned char *buff)

【入力条件】 head 画像の線形ﾘｽﾄ．全画像の高さは同じ

sizex, sizey 画像の縦･横のサイズ(int 値)を格納するﾎﾟｲﾝﾀ

【出力条件】 sizex, sizey 画像サイズを sizex, sizey に入れる

buff この関数内で，画像ｴﾘｱを確保して，画像を格納する

【アルゴリズム】 (1) ﾘｽﾄ(head)内の画像の横サイズの合計値を求める．

(2) 画像の縦サイズを求める（先頭画像のサイズを使う）．

(3) 結合画像用の画像ﾊﾞｯﾌｧを確保する．

(4) ﾍｯﾀﾞの出力，画像本体の出力を行う．

2.2.4. pgm **画像ライブラリ**

●画像ﾍｯﾀﾞﾛｰﾄﾞ

【関数名】 void pgm_scan_header(FILE *fp, int *sizex, int *sizey)

【入力条件】 fp 画像ﾌｧｲﾙを読み込みｵｰﾌﾟﾝしたもの

【出力条件】 sizex, sizey 画像の横･縦のサイズ（ﾋﾟｸｾﾙ数）

●画像本体ﾛｰﾄﾞ

【関数名】 void pgm_load_body(FILE *fp, unsigned char *buff, int sizex, int sizey)

【入力条件】 fp 画像ﾌｧｲﾙを読み込みｵｰﾌﾟﾝしたもの

buff 画像の格納先．十分なﾒﾓﾘが確保済みであること．

sizex, sizey 画像の横･縦のサイズ

【出力条件】 buff 画像を格納

●画像セーブ

【関数名】 void pgm_save_image(FILE *fp, int sizex, int sizey, unsigned char *buff)

【入力条件】	fp	画像ファイルを書き込みオープンしたもの（stdout 可）
	sizex, sizey	画像の横･縦のサイズ
	buff	出力画像のピクセルデータ
【出力条件】	（ fp で示すファイル（または標準出力）に画像ファイルのヘッダ部, および本体を出力）	

2.2.5. 線形リスト操作ライブラリ

●リストノードの生成

【関数名】 Node *list_make_node(unsigned char *buff, int sizex, int sizey);

【入力条件】	buff	画像データ. ポインタを渡す.
	sizex, sizey	画像の横, 縦のサイズ
【出力条件】	Node *	新しく作成したノードへのポインタ.

●リストノードの追加

【関数名】 void list_push_node(Node *head, Node *data);

【入力条件】	head	画像のリストの先頭ノード（ダミーノード）
	data	挿入する画像のノード
【出力条件】	head	head の示すノードが data になり，data 以降に head の元々示していたノードが続く.

2.3. 物理データ設計

2.3.1. pgm 画像ファイル

pgm ファイルは, 次の構造とする.

P5　横サイズ　縦サイズ　最大輝度〈改行〉[*1]
輝度値①輝度値②輝度値③…

ここで, 輝度値は 0 〜 255 の整数値である. よって, バイナリファイルとなる. 輝度値と画像の座標との対応は, 輝度値①が(0,0)の座標で, 画像の左上となる. 輝度値②は(1,0)の座標の輝度値で, 輝度値①の真横に位置する.

2.3.2. リストファイル

画像のファイル名を列挙したもの.　fopen　で直接使うので, ディレクトリのパスを含めることを許す. 以下に例を示す.

img01.pgm
img02.pgm
img03.pgm

また, 最後のファイル名の末尾に改行コードが必ず存在すること.

2.3.3. 画像格納ノード（Node）

ノードは, 画像の縦･横のサイズ, および, 画像のピクセルデータへのポインタを保持する. ピクセルデータは, 必要なだけ malloc　により確保する. ピクセル値は, 0 〜 255 であるので, unsigned char 型となる. また, リスト構造を形成するため, 次のノードへのポインタも有する.

```
typedef struct inode {
    int sizex, sizey;
    unsigned char *buff;
    struct inode *next;
} Node;
```

*1 各項目の間はスペースが入る. 一般の pgm ファイルでは, スペース以外に改行コードが入る. また, 行頭に#　があるならば, その行は読みとばす. 本ソフトウエアでは, スペースが入る場合にのみ対応する.