

画像処理実験支援用データ管理ライブラリの開発

1F-8

松尾治夫 横矢直和
奈良先端科学技術大学院大学

1 はじめに

画像データのファイルフォーマットは各研究分野で標準化が行なわれている[1, 2, 3]。しかし、研究に用いられる画像のデータ形式は様々であり、研究分野間のデータ交換はあまり行なわれていない。これは画像データ本体のみでなくそれに付随する情報も研究分野によって異なることに起因する。本稿では、画像処理の研究者が様々な形式の画像データを容易に管理、処理、交換できる汎用画像データフォーマットを提案し、これに基づくデータ管理ライブラリについて述べる。

2 設計思想

画像処理システムは一般に以下の機能を持つ。

- ユーザコマンドの解釈
- 画像入出力装置の制御
- 画像処理アルゴリズムの実行
- 画像データ・ファイルの管理

現在、開発中のライブラリは、主に画像処理アルゴリズムの実装を支援するため、画像データ・ファイルの管理を行うものである。開発は UNIX¹上の C 言語で行なった。このライブラリには以下のような特徴がある。

- 画像処理の研究者を対象とし、画像データのファイル入出力を容易に行なうライブラリを提供。
- ユーザがヘッダファイルで画像などのデータ構造を定義可能。
- ヘッダファイルに記述した画像データやそれに付随する情報をユーザプログラムから参照可能。

Data Management Library for Image Processing Experiment

Haruo Matsuo and Naokazu Yokoya
Nara Institute of Science and Technology
8916-5, Takayama, Ikoma, Nara, 630-01, Japan
¹UNIX は、X/Open Company Limited の登録商標です。

- ppm, pgm や IVBox [4] 等の既存の非圧縮型のデータファイルフォーマットに対応可能。

処理の対象となるデータ型として次のものを想定する。

- 文字列型: 画像の名前や、日付、コメントなど
- 数値型: 画像のサイズや、縦横比など
- 配列型: 通常の濃淡画像、距離画像、ボリウムデータなど n 次元配列で表されるデータ
- 構造体型: 座標データとその位置の輝度情報など構造体で記述可能なデータ

3 データフォーマット

3.1 データを記述するファイルの種類

ファイルは磁気ディスク上にあることを前提としており、既存のフォーマットにも対応できるように、ヘッダファイルとデータ本体（データファイル）を分けることができる。本ライブラリが使用するファイルには次のものがある。

ヘッダファイル: データ構造の定義を行ない、画像の属性、処理に用いる情報の記述を行なうファイル。データ構造はフォーマットファイル、データ本体はデータファイルに分けて指定することもできる。1つのヘッダファイルに複数の画像データを記述することができる。

フォーマットファイル: データ構造の定義を行なうファイル。複数のヘッダファイルで同じフォーマットを用いる時に記述し、ヘッダファイルから参照する。

データファイル: 画像データなどを格納するファイル。テキスト形式とバイナリ形式がある。ヘッダファイルで格納領域を指定して読まれる。

3.2 データ構造の定義とデータの記述

データ構造の定義やデータの記述には表1のコマンドを用いる。また、データ構造を定義する際は表2に示すデータ型を使用する。

例えば、RGB 各 1 byte で 1 画素が 3 byte のカラー画像の場合、画像データファイルは図 1 のようになり、そのデータ構造をフォーマットファイルに記述すると図 2 のようになる。この場合のヘッダファイルを図 3 に示す。

表 1: データ構造定義のためのコマンド

命令	意味
macro	マクロ定義
defkey	データ構造の定義
setkey	データ格納領域の確保、初期値の設定
begin	データのはじまり
end	データの終り
form	フォーマットファイルの指定
bload	バイナリ形式データファイルの指定
load	テキスト形式データファイルの指定

表 2: データ型

型名	意味
char	8 bit signed int
uchar	8 bit unsigned int
short	16 bit signed int
ushort	16 bit unsigned int
int	32 bit signed int
uint	32 bit unsigned int
float	32 bit float
double	64 bit float
bppXX	XX = 1~32 bit unsigned int

4 データ管理ライブラリ

本ライブラリがユーザに提供する関数の一部を表 3 に示す。

表 3: ライブラリの関数

関数名	機能
ILinit	データ格納領域の初期化
ILload	ファイルの読み込み
ILstore	ファイルの書き出し
ILkeypnt	データ格納領域のアドレスの取得
ILcom	データ構造の定義

フォーマットファイルに記述したデータ構造は C の構造体宣言と 1 対 1 に対応する。例えば、図 1 の画像データを読み込んで、輝度情報の格納域のアドレスを得る C のプログラムは図 4 のようになる。プログラム中では関数 ILload でヘッダファイルを指定してデータファイルを読み込み、関数 ILkeypnt でデータ格納領域 rgb24 のアドレスを得ている。

5 まとめ

データ構造をユーザがヘッダファイル中で定義できるようにしたことで様々な画像フォーマットに対応できるようになった。また、データ構造をフォー

マットファイルで管理するので、ある研究分野で作られたフォーマットファイルを他の研究分野で利用でき、研究分野間のデータ交換が容易に行なえるようになった。

現在は圧縮されたデータについて考慮していないので、今後は圧縮されたデータに対応できるよう拡張して行く予定である。

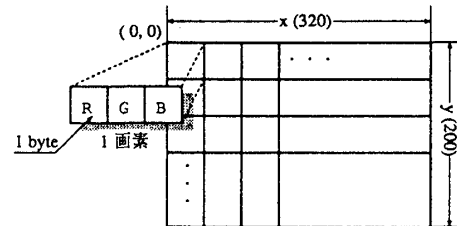


図 1: 画像データファイルのデータ格納例 (imagefile)

```
defkey RGB24
  uchar r
  uchar g
  uchar b;
```

図 2: フォーマットファイルの例 (formatfile)

```
form formatfile;
setkey RGB24 rgb24[200][320];
begin
  rgb24: bload imagefile;
end
```

図 3: ヘッダファイルの例 (headerfile)

```
ILKeyword *kw;
struct {
  unsigned char r;
  unsigned char g;
  unsigned char b;
} *rgb24;

kw = ILinit();
ILload(kw, "headerfile", 0);
rgb24 = ILkeypnt(kw, "rgb24");
```

図 4: データ格納領域のアドレスを得るプログラムの例

参考文献

- [1] J.L. Mundy, R. Welty, and K. Price (and the IUE Committee): "The Image Understanding Environment: Data exchange", *Proc. Image Understanding Workshop*, pp. 301-319, 1993.
- [2] NASA/Science Office of Standards and Technology: "Definition of Flexible Image Transport System (FITS)", NASA Goddard Space Flight Center, 1993.
- [3] コンピュータビジョン研究連絡会: "拡張標準画像データフォーマット", 情報処理, Vol. 30, No.1, pp. 50-57, 1989.
- [4] 山崎, 坂上, 横矢, 山本, 松原, 加藤: "UNIX 環境下での画像処理支援ツール IV-Box", 映像情報 (I), Vol. 20, pp. 33-39, 1990/1.