

データベース操作ツールキットの機能

宮本 崇夫

(株)リコー ソフトウェア事業部

本稿では、イメージデータベースを構築する手法と、その上でのイメージデータ処理方法について論じる。

イメージデータにはデバイスに依存したデータ形式や業界標準のファイル形式が多種存在する。そのため、イメージを扱うためには、デバイスに依存した専用の手続きが必要である。データ形式間の変換手続き、およびイメージの入出力デバイスの特性をデータベースに格納することによって、必要な形式のデータを得ることができるようになる。

また、イメージデータといった視覚的なデータを扱う場合、ユーザーインターフェースが重要になる。X11 でユーザーインターフェースを構築する例を述べる。

Toolkit for Database Manipulations

Takao Miyamoto

Ricoh Software Division

1-1-17, Koishikawa, Bukyo-ku Tokyo 112, Japan

This paper describes toolkits and techniques for the construction of an image database. Generally, image handling is harder than text handling for the following two reasons. The first one is the large amount of data to be handled. The second one is due to the variety of image file formats and types. Therefore, the troublesome conversion between file formats and types is required. To solve this problem, we define new record types for the image, develop the image handling toolkit, and store the toolkit procedures as a part of the image database. These capabilities developed on the X multi-window system are also described.

1 はじめに

近年の計算機の高性能化、各種周辺機器の進歩により、扱うことのできる情報量が飛躍的に増え、様々なデータを扱えるようになってきた。これらの多種多様なデータ及び入出力機に対応するためマルチメディアに関する研究、開発がさかんに行なわれている。

本稿ではイメージデータベース構築のための手法と、その上でのイメージ処理方式について論じる。

イメージデータの入出力機器としてはスキャナー、ファクシミリ、ビデオカメラ、レーザープリンター、ビットマップディスプレイなどがあるが、これらを用いてイメージを取り込んだり、イメージを表示・印刷したりする際には、そのデバイスに依存した専用の手続きが必要となってくる。イメージを扱うデバイスの特性は、通常のテキストデータを扱うデバイスに比較して多様であるため、以下のような拡張が必要とされる。

1. データ型の拡張
2. イメージデータに対する関数や演算の追加・変更
3. データに適したファイル構造やアルゴリズムの追加

本稿では、上記1,2に対応する「イメージ型、画像型の導入」と「メディアデータ形式間の変換手続き」について考察し、その手続き自体をイメージデータベースの一部として格納することを考える。

2章ではイメージデータの処理の特徴について、3章ではイメージデータベースの構築に関して、4章はイメージデータベースの操作及びXウィンドーを用いたユーザーインターフェースについて論じる。

2 イメージ処理の特徴

ここでは、イメージデータベースを扱う上で必要とされるイメージ処理について述べる。

2.1 イメージファイルの形式とその変換

イメージデータを扱うアプリケーションを作成する際、業界標準として普及しているファイル形式がいくつ

かある。こうした標準的形式間の変換手続きは独自で作成しなくても、パブリックドメインとして、あるいは商品として入手可能である。

実用システムでは性能向上のため、特殊なファイル形式を定義し、それに基づいて入出力機器制御プログラムを作成しなくてはならないことが少なからずある。しかし、このような場合も、その特殊ファイル形式から業界標準ファイル形式の1つへの変換手続きを作成しておけば、他の各種形式への変換が可能となる。

以下に代表的な業界標準ファイル形式を示す。

• TIFF(Tag Image File Format)

従来のファイル形式の多くの編成が固定的であることに比較して、TIFFはその編成が柔軟であるという特徴を持つ。TIFFでは、多種のデータ圧縮方法、複数のイメージ解像度、バイナリ/グレースケール/カラーのような異なったイメージのタイプをサポートし、これらの記述を追加、更新することも可能にしている[1]。

• xwd(X Window Dump format)

X ウィンドーシステムの標準ファイル形式で、その編成は固定的である。X ウィンドーシステムでは、xwd形式のファイルの操作や表示用ライブラリを提供しているため、X ウィンドーを通してのイメージ処理を考えた場合有効である。

• SUN ラスター

SUN Microsystems, Inc.の設定した標準フォーマットで、SUN ワークステーション上ではこの形式のラスターイメージの操作ライブラリ、コマンドが用意されている。イメージを記述するための必要最小限の機能を持つ。SUN ワークステーションのユーザーが多いのでSUN ラスターのアプリケーション蓄積も多いといえる。

2.2 イメージデータの圧縮・伸張

イメージデータは長大であるので、格納する際にはデータの圧縮を行なうのが一般的である。

例えばファクシミリで取り込んだデータをラスターファイルとして格納する場合、送られてきたままの圧縮形で格納した方が記憶容量が少なくすむ。(圧縮しな

ければ、A4、白黒2値、200dpi(8dot/mm)で約500KB程度である。)

標準的な圧縮/伸張アルゴリズムとしては以下のようなものがある。

- MH 符合(Modified Hoffman Code)
ランレングス(1や0の連続する数)と符合の変化をコード化することによって圧縮する。符合の変化とコードとの対応は対応表によって定義される。UNIX 標準コマンドの pack/unpack はこの手法を用いている。
- MR 符合(Modified Read Code)
注目走査線と一つ前の走査線との差分を符合化する。CCITT 標準原稿平均でほぼ20倍の圧縮効率である。回線ノイズなどの原因によるエラー波及を防ぐためkラインごとに全ラインMH符合化を用いる。上のMH法とともに、G3ファクシミリで採用されている。
- MMR 符合(Modified MR)
k=∞のMR。G4ファクシミリで用いられている圧縮法である。

2.3 イメージデータの操作

各デバイス間でのデータ転送、ならびにデータベースへの格納を考える際、各メディア間でのデータ形式の変換、圧縮・伸張の他に、ある種のイメージデータ操作が必要なことがある。

例えば、白黒のプリンタにカラーイメージを出力する場合を考えてみる。まず、カラーRGB値から輝度への変換を行い、次にこの輝度から白黒イメージを作成する。その際、組織ディザ法や誤差拡散法によって、2値化するのが普通である。2値化のアルゴリズムはこの2種の他にも多種にわたっているため、状況にあった技法を選択させるため、システム側ではユーザーを補佐したり、デフォルトの手法を提示するなりして、何らかのユーザー支援をすべきであろう。

この種のイメージデータ操作として以下のようなものがある。

- ラスターイメージの変倍

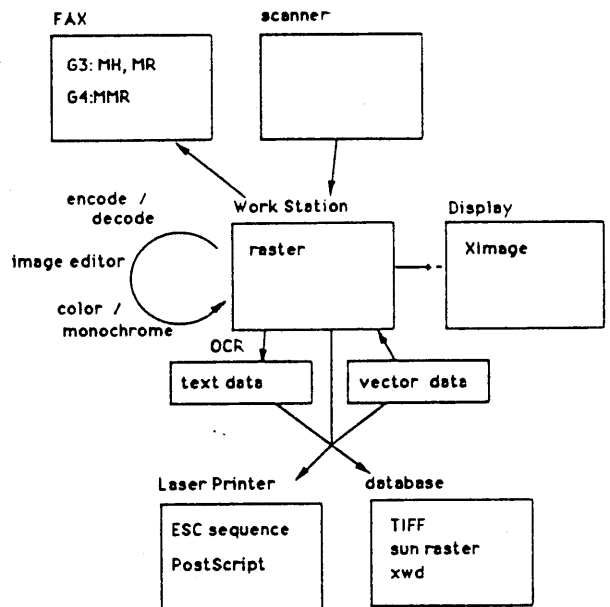


図1: イメージデータの流れ

- ラスターイメージの回転
- ラスター/ベクター変換
- カラー/白黒変換
- カラー/カラー変換
- OCRによる文字認識
- 線認識によるフォーム(矩形領域)の抽出

カラー/カラー変換とは、デバイスに依存するカラーマップテーブルの深さの違いなどによるカラーイメージデータ形式間の変換を示す。

以上、2.1から2.3に記述した形式の変換や各種イメージの操作、デバイスを含めたイメージデータ処理の流れを図1に示した。

3 イメージデータベースの構築

この章では、イメージデータベース構築手法について論じる。イメージデータベースを構築するにあたり、

(株) リコーが開発した、UNIX 上で作動する汎用データベース管理システム「G-BASE」を用いた。G-BASE は、無制限可変長を含む豊富なデータ型を備え、リンクによりレコード間の関連を表すことが可能な拡張リレーショナル型の DBMS である[2][3]。

3.1 イメージ・データ型

イメージ型、画像型といったレコード型を定義することにより、メディアごとに異なる形式をもつデータをデータ型として統一的に扱うことが可能となる。以下に、G-BASE 上で定義したイメージのレコード型を示す。属性として最小限のイメージ形式記述要素しか含んでいないが、他の特性の追加もそのレコード型を定義し、イメージレコードとイメージ特性レコードとの間にリンクを張ることによって実現できる。

イメージデータベースで検索を考える際、一つのデータに対して検索キーとして、複数のテキストデータを持たせることが必要であるが、このイメージ検索キーについても同様に、イメージと検索キーとの間にリンクを張ることによって、多対多の関連を構築することができる。

- イメージ型のデータ型

```
record IMAGE (
  id          integer(4),      /* id */
  width       integer(4),      /* 横幅(dot) */
  height      integer(4),      /* 高さ(dot) */
  format_type integer(2),      /* フォーマット */
  depth       integer(2),      /* 色の深さ */
  encode_type character(256), /* 圧縮のタイプ */
  data        variable        /* ラスター */
                character with null code "bin"
)
```

"variable character" とは無制限可変長文字列型を表す。

- デバイスのデータベース化

この他、各種イメージ入力機器の属性もデータベース化し、統一的に扱えるようにする。例えば、X ウィンドーシステム環境下のビットマップディスプレイ装置のレコード型を以下のように定義した。

```
record DISPLAY (
  id          integer(4),      /* id */
  display_name variable       /* ディスプレイ名 */
                character(256),
  width       integer(4),      /* 横幅(dot) */
  height      integer(4),      /* 高さ(dot) */
  depth       integer(2),      /* 色の深さ(1,8,24) */
  color_type  integer(2)      /* カラーマップ id */
)
```

Xでは、サーバーのディスプレイ名は、ホスト名:"ディスプレイ番号","スクリーン番号"のように表記される。これは1台のディスプレイ(ワークステーションと呼ぶこともある)が複数のスクリーン(CRT)をもつこともあり、また、1台のホストに複数のディスプレイを接続することも可能であるからである。その他、ディスプレイの属性として、色やグレイスケールを表現するためのフレームメモリの深さ(色の深さ)、その値からRGB値(あるいはグレイスケール値)に変換するためのカラーマップがある。

上記のようなデバイスに依存した情報は、適宜、異種デバイス間のイメージ変換に利用される。これにより、ユーザーはデバイス依存部を意識することなく変換を行なえるようになる。

- 変換手続きのデータベース化

イメージデータに対応する演算の追加や更新の機能は、メディア形式ごとに必要とされる演算が異なるため、あらかじめデータベースに登録しておくことは不可能である。よって、新しいイメージデータ形式を定義する際に、異なるメディア形式間の交換についても、その各変換ごとに交換ルール/手続きを登録する必要がある。この変換は、従来のDBMSが行っていた数値型の変換や、文字コードの変換といった単純な内容ではなく、デー

タの性質に応じた圧縮/伸張、各種の処理手順が含まれる。

```
record OBJECT (  
  id          integer(4), /* id */  
  func_name   variable /* 関数名 */  
              character(256),  
  machine_type variable /* マシンタイプ */  
              character(256),  
  cpu_type    variable /* CPU のタイプ */  
              character(256),  
  os_type     variable /* OS のタイプ */  
              character(256)  
  size        integer(4) /* バイナリの大きさ */  
  func        variable /* 実体 */  
              character with null code "bin"  
)
```

8plane のカラーデータを 1plane の白黒のデータに変換する例では、以下のような値が代入される。

```
func_name    = 8to1  
machine_type = Sun3/60  
cpu_type     = mc680020  
os_type      = SUNOS 4.0
```

関数は動作する条件として、以上のように、マシン名と CPU と OS が決まればよい。

- トランスレーションマネージャー
データ変換を管理するプロセスをトランスレーションマネージャーと呼ぶ。トランスレーションマネージャーに変換元と変換先のファイル形式を与えると、その間のデータ変換手続き列を返す。複数の変換技法がある場合、デフォルトではもっとも頻繁に使われているものをトランスレーションマネージャーが自動的に選択するが、対話的にユーザーが選択することも可能である。

4 データベースの操作

4.1 データベースの操作例

以上のようにデータの型を定め、イメージデータベースを構築する。例として、『あるイメージデータを検索し、そのデータを指定したディスプレイに表示する。』と、いった操作手順を考える。処理の流れは以下のように表現される。

1. データの検索

キー検索により、該当するイメージデータオカレンスが得られる。リンクをたどることにより、そのデータの格納されている形式の情報なども得ることができる。

2. デバイスの検索

指定のディスプレイのデバイスの特性をデータベースより検索する。

3. データの形式の変換

以上の 1,2 で得られたデータをトランスレーションマネージャーに与えると変換に必要な処理手続きが提示される。イメージデータはその処理手続きの実行により、変換され、指定ディスプレイに表示される。

4.2 グラフィカルユーザーインターフェース

上記の操作のためのグラフィカルユーザーインターフェースを X ウィンドーを用いて試作し、検討してみた。図 2 に示すように、あるウィンドー上にネットワーク上の機器構成が表示されている。各アイコンはネットワーク上の利用可能なイメージの入出力機器であり、ファクシミリ、スキャナー、レーザープリンタ、データベース(外部記憶装置)、ビットマップディスプレイなどに対応している。この画面上でアイコンを用いることによりユーザーはイメージの操作を行なうことができる。

例えば、スキャナーからデータを読み込み、データベースに格納し、プリンタに出力する場合の操作の流れを示す。

1. データの読みとり

ユーザーがスキャナーのアイコンをクリックする

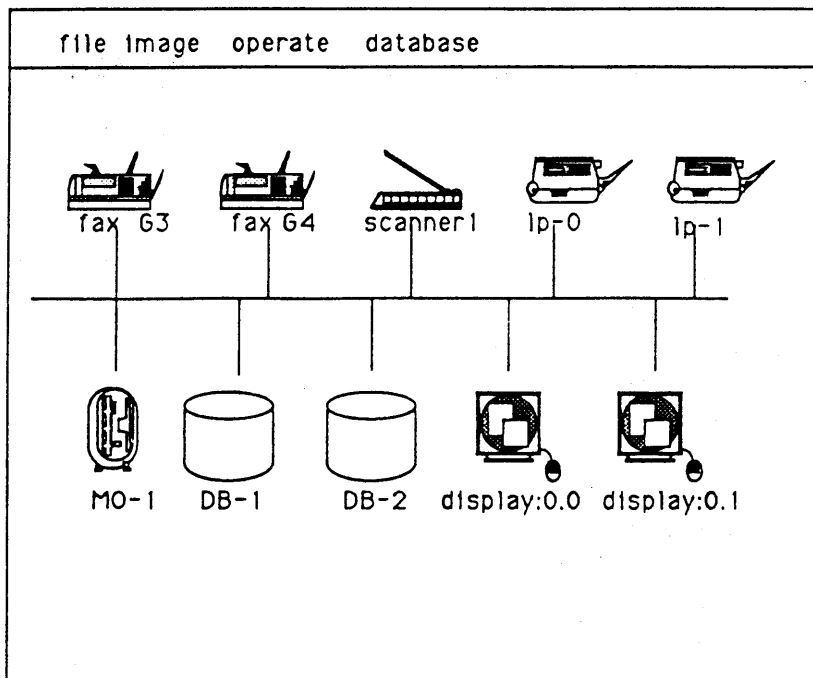


図 2: イメージデータベース操作画面

と、操作メニューが画面上にポップアップされる。スキャナー読み込みの属性値をセットした後、スキャニングが実行される。実行後、データがアイコン化され、画面上におかれる。この時のフォーマットはスキャナーで読みとったそのままのフォーマットである。

2. データのプレビュー

イメージデータの内容とその特性を見るためには、図 3 のようなプレビューアを用いる。そのイメージ自体とそのサイズなどのイメージの持つ属性値、並びに検索キーなどのテキストのデータが表示されている。

3. データベースへの登録

画面上にアイコン表示されているイメージデータをデータベースに登録するには、データアイコンを選び、次に格納先のデータベースアイコンを選ぶ。データベース操作メニューから、「登録」のコマンドを選択すると、キー検索などの属性入力を促すので、ユーザーはそれらを指定した後、登録を実行

する。

4. プリンター出力

白黒のレーザービームプリンターを対象とした場合、一般にカラー画像を 2 値化する際、ハーフトーニングする必要がある。デフォルトはディザ法であるが、メニューによって、イメージ処理を選択し、その処理を手続きデータベースから検索することにより、イメージに適用することができる。

5 まとめ

イメージデータベースを構築するためには、イメージに対応したデータ型の定義ができることと、イメージデータに対する関数や演算の演算をデータベースが自由に使えるようになる必要がある。イメージデータは多種多様なデバイスと形式をもつので、それぞれに対して変換、圧縮・伸張、その他の操作が必要となる。

そこで、本稿では G-BASE 上にイメージデータ型を定義し、デバイス特性、メディアデータ形式間の変換手続き自体もデータベース化することにより、統一的にイ

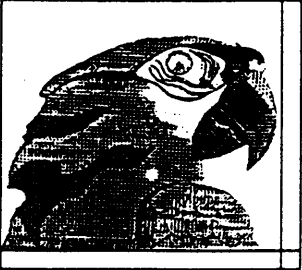
file image edit database	
blue bird	width 200
	height 300
	depth 8
	encode dither
Message	please input retrieval key

図 3: イメージの描画

メージを扱える環境の構築を試みた。イメージデータベースの場合、グラフィカルユーザーインターフェースの質が操作の簡便性に大きく影響するので我々は今後とも、DBMS 操作用グラフィカルユーザーインターフェースの研究を続けて、ユーザーにより快適な環境を提供していきたい。

謝辞

本稿をまとめるに当たり、(株)リコーソフトウェア事業部第2研究室の白田氏、飯沢氏には多くの助言指導をいただいた。また、第2研究室、第2開発室のメンバー各位の協力に対し、感謝の意を表す。

参考文献

- [1] 日本電気(株)編:「Software Library MS-WINDOWS エクステンションズ」,1988.
- [2] Kunii,H.S.: Graph Data Language: A High level Access-Path Oriented Language, Ph.D. Dissertation, the University of Texas at Austin, 1983.
- [3] (株)リコー編:「G-BASE システムマニュアル」,1989.
- [4] 飯沢、白田:「マルチメディア・オブジェクト指向によるデータベースの構築とその例」,技研情報センターセミナー資料 (1990.2.22).