

# 画像データの保存処理に関する研究

白 崎 智 義\*

## Study on Preservation Processing of the Image Data

Chiyoshi SHIRASAKI

The image of the computer screen is stored in the preservation medium of the floppy disk as the data. Then, this image data is taken out of the medium and it is displayed on the screen. At this time, it is natural that the same image as the original picture image is displayed. However, the phenomenon of differing from the original picture image might happen some cases.

To investigate this cause, an examination has been made in the storage of memory in the medium. As a result, it has been understood that there are two kinds of form of preservation structure in the memory. It has been understood that this is the cause of the above-mentioned image destruction.

The study has indicated that it is necessary to note the usage of various instructions prepared for the image processing.

### 1. まえがき

コンピュータ画面の画像をデータとしてフロッピーディスクの保存媒体にいったん格納してから、この画像データを取り出して画面に表示すると普通なら格納する前の原画像そのものが表示される筈なのに、何故か原画像とは異なったわけのわからぬ画像が出ることもある。

この原因を探る為に、媒体内のメモリの保存状態を調べる研究をした。この結果、メモリの保存の形式が2種類あることが分かった。そして、これが上述の画像破壊の原因につながっていることも分かった。

つまり、画像処理用として用意されている各種命令の取扱いに注意を要することを痛感した。

今回の研究で使用した OSはDOSで、言語はTurbo-Cである。尚、BASICの場合におけるフロッピーディスクのデータの保存実態については、既に文献<sup>(1)(2)</sup>で報告してある。

### 2. 画像データ

画像処理による画像データの状態変化の様子を詳細に観察する為には、それなりの基礎的要素だけを含む簡単なデータを利用するのがよい。ここで、CCDカメラ等による外界データの読み込みが考えられるが、この場合はどうしても雑音が入りやすく基素としての研究対象にはなりにくい。これは基礎研究終了の段階でより充実した内容の濃い研究用画像処理のデータとして利用する。

そこで、雑音が全く入っていない任意形状の画像データを作成することにした。このデータを使って、フロッピーディスクへの入出力に関連するコマンドと画像が正しく画面に表示されるかどうかとの関連を実験的に研究した。

フロッピーディスクへの入出力関連作業は次の通りである。

---

\* 電気工学科

- (1) データを作成する
- (2) フロッピーディスクから画像データファイルを取り込む
- (3) 画面データをフロッピーディスクに保存する
- (4) 画面上における表示と抽出

上の各場合について行った研究内容は次の通りある。

(1) データを作成する

この作成にあたっては、(a) DOS のコマンドである symdeb を利用して直接メモリ内に画像データを書き込む方法と、(b) キーボードのキーを利用して画面上で直接画像を描画する 2通りの方法を使った。

(a) symdeb の利用による方法

1. 予め、作りたい画像データの形状を方眼紙上で設計して模式図を作る。
2. この模式図を見て、図上の各点に対応するメモリ内の各ドットの位置を 16 進数で計算し、1 次元配列状に次々と濃度値を書き込んでいく。
3. 画像の形状を縦横  $n \times n$  の大きさの正方形とし、その領域の大きさを 16 進数に変換して CX レジスターに設定する。
4. 出来たデータにファイル名を付けて、フロッピーディスクに格納する。

例えば、模式図による画素の配置が  $64 \times 64$  行列状になっている場合、上の操作を実施する操作は次の様になる。

カレントドライブ A において、A > symdeb  
とする。

これに引続き、次の各例で示すように、[ ] の中に書き添えてある コマンド を使うことで、それぞれに応じた命令が実行出来る。

1. 各セグメントや各レジスターの現状のデータを確認する。、[-r           ]
2. 例えば、 $n=64$  の場合は、CX レジスターに 1000H を設定する。[-rcx 1000 ]
3. 例えば、メモリ 300 番地から画像データを書き入れていく。[-e300       ]
4. 作成した画像データにファイル名(例、abc.dat)を与える。[-n abc.dat]
5. 作成した画像データをフロッピーディスクに保存する。[-w           ]

尚、1 画素の濃度値は 1 バイト長さの 16 進数値で構成されていて、上位 4 ビットで色付けをし、8 ビット全長で濃度を表わしている。

(注) symdeb のこの他への利用法

7. フロッピーディスクの内部に格納されている画像データの情報を確認するには、

-l100 1 5 1  
-d100

とし、このダンプによって、1 ファイル毎に (表 1) の様な形式で情報が 2 行に纏められているのが分かる。ここで、l はルートディレクトリから load する為の コマンド であり、これに続く 100 はユーザー領域の最初の番地、次ぎの 1 はドライブ番号で普通にいう B ドライブ のことであり、5 はファイル関係の情報を見る為の数値であり、最後の 1 は取り出すセクタの個数である。

(表 1) ファイル情報

00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
ファイル名								識別子		属性					
						作成時刻		〃年月日		クラスターNO.		ファイルの大きさ			

ここで、参考までに研究で作成した "EXAMP.DAT" ファイルの場合を (表 2) に示す。  
(表 1) との対応関係から、(表 2) に保存されているファイル名、属性、作成時刻、  
作成年月日、クラスタ NO、ファイルの大きさに関する情報が分かる。

(表 2) "EXAMP.DAT" ファイルの情報

45	58	41	4D	50	20	20	20	44	41	54	20	00	00	00	00
				※1						※2					
00	00	00	00	00	00	10	63	82	21	02	00	00	10	00	00
						※3		※4		※5			※6		

※1 により、ファイル名が EXAMP で、識別子が DAT である。

※2 により、属性が 20 だから普通のファイルとして扱う。

※3 により、10 63 を入れ替えて 63 10 とし、これをそのまま 2 進数に表示替えし  
て、

0110 0011 0001 0000

とし、さらにこれを 01100 011000 10000

のように組替えると、10 進数で 12, 24, 16 の並びになる。結局、作成した時刻は、  
12 時 24 分 16 秒となる。

※4 により、82 21 を入れ替えて 21 82 とし、これをそのまま 2 進数に表示替えし  
て、

0010 0001 1000 0010

とし、さらにこれを 0010000 1100 00010

のように組替えると、10 進数で 16, 12, 2 の並びになる。結局、作成した年月日は、  
始めに、1980+16=1996 の年計算をして、1996 年 12 月 2 日となる。

※5 より、ファイルを保管する最初のクラスタ番地は、上位と下位を入れ替えて 02H と  
なるから、これに 16 進 9H を加算して得た数値 0bH を使って、

-1100 1 0b 1

-d100

とすることで、内部メモリの配列状況が直接自分の目で観察することが出来る。

※6 より、ファイルの大きさは、4 バイト表示だから、まず 2 バイト毎に上位と下  
位のバイト交換をし、続いて各バイト毎に上位と下位を入れ替えると、1000H となる。  
これを 10 進数になおすと 4096 になる。

#### イ. フロッピーディスクの媒体識別

-1100 1 1 1

-d100

で FAT テーブルを表示し、最初の 1 バイトが例えば FEH ならば 2HD, FBH ならば 2DD (640KB) の  
媒体であると識別出来る。

#### (b) キーボードによる方法

画面上で、キーボードのキー操作により設計模式図を見ながら直接画像を描画する。  
そして、この画像をデータとしてフロッピーディスクに格納する。

#### (2) フロッピーディスク から画像のデータファイルを取り込む

フロッピーディスク に格納されているファイル形式のデータファイルを取り込むには、予め、

fp=fopen(ファイル名, "rb");

の設定しておく。

次いで、

```
fread(buffer, xsize, ysize, fp);
```

により、xsize長さ分のデータをbufferに入れる作業を、下のように while文（一例）で連続して ysize回繰り返す。

```
while(fread(buffer, xsize, ysize, fp) == NULL){
```

ここで、xsize, ysizeは画像領域の横、縦の長さを表している。尚、bufferは予め次のように2次元配列の宣言をしておく。

```
unsigned char buffer[ysize][xsize];
```

(3) 画面データを7ビット・ディスクに保存する

7ビット・ディスクに画像データを格納するには、予め

```
fp=fopen(ファイル名, "wb");
```

の設定をする。そして、

```
fwrite(buffer, xsize, ysize, fp);
```

により、既にbufferに入っている画素データを 7ビット・ディスクに格納する。

(4) 画面上における表示と抽出

7. bufferに内蔵されている画像データを、(x1,y1)を左上端、(x2,y2)を右下端の座標とする領域に表示するには、次の命令が利用出来る。

```
display(buffer, x1, y1, x2, y2);
```

4. (X1,Y1)-(X2,Y2)の方形領域の画像データを面積単位で一度に取り出すには、

```
getimage(x1, y1, x2, y2, buffer);
```

として、データ全部をbufferに格納する。

これを同一画面内の他の場所に再表示するには、

```
putimage(x1, y1, buffer, COPY_PUT);
```

とする。ここで、COPY\_PUTはコピーの機能をもつ描画演算である。

画像内のある部分領域を設定する作業は、マウスまたはキーボードのキーを利用する。

ウ. 画面上の1個の画素(座標 x,y)をgetpixel()で抽出して、これを別の場所に再表示するには、

```
dat=getpixel(x, y);
```

の後、

```
putpixel(x1, y1, dat);
```

とする。ここで、dat は取り出した色、濃度値のデータを受け取る変数である。連続して、領域の画素すべてについて実施するには次のようにする。尚、xposi1,yposi1 は最初に取り出す画素の位置座標、xposi2,yposi2 は最初に置く画素の位置座標を表す。

```
for (i=0; <xsize; i++){
    for (j=0; <ysize; j++){
        dat=getpixel(xposi1+x, yposi1+y);
        putpixel(xposi2+x, yposi2+y, dat);
    }
}
```

(5) 各命令の組合せ

画像入力側の各命令と画像出力側の各命令の対応関係は（表3）に示す通りである。各命令の（ ）内には、必要に応じて適当なパラメータを書き込めばよい。

(表3) 各命令の組合せ

画像入力側		画像出力側	
1	フロッピー・ディスクから取り出すには、 fread(                    );	3	フロッピー・ディスクに保存するには、 fwrite(                    );
2	画像のデータを取り込むには、 getimage(                    ); getpixel(                    );	4	画像を表示するには、 display(                    ); putimage(                    ); putpixel(                    );

### 3. 実験

#### (1) データの保存形式

画像データをメモリに格納するとき、使う命令によって保存状態に次の2通りがある。

- (a) 方眼紙で設計した絵図の配置状態でメモリが構成されている。
- (b) 上とは異なる状態でメモリが構成されている。

このことを確認する為に、次の実験をした。

まず、幅2ドットで長さ8ドットの短い横線を画面上にFFデータで描画出来るようにする為、64×64の2次元配列のメモリ作成に便利な(図1)のような模式図を方眼紙上に作った。

		X <sub>1</sub>										X <sub>2</sub>									
		0	1	2	3	4	5	6	7	8	9	10	.	.	.	.	61	62	63		
y <sub>1</sub>	0													~							
	1													~							
y <sub>2</sub>	2			P	●	●	●	●	●	●	●	●	Q	~							
	3			R	●	●	●	●	●	●	●	●	S								
	.																				
	.	~	~	~	~	~	~	~	~	~	~	~	~	省略			~	~	~		
	.																				
	62													~							
	63													~							

(図1) 模式図

線分の始点にあたる点Pの画素データを格納するメモリ番地を求めるには、まず、模式図で原点(0,0)から数えた点Pまでの長さ(Lp)を、

$$Lp = 64 * y_1 + (x_1 + 1)$$

で求める。

ところで、メモリデータの配列が1次元であるから、点Pのメモリ位置(Mp)はLpの数値を16進数に変換し、これに100Hを加えたものである。

よって、今の例では、x<sub>1</sub>=3, y<sub>1</sub>=2 だから、

$$Lp = 64 * 2 + 4 = 132$$

これを16進数になおして、100Hを加えると、 $Mp = 184H$   
 になる。また、線の終点Q点についても同様に考えて、  
 $Lq = 64 * 2 + 11 = 139 \quad \therefore Mq = 18BH$   
 になる。そして、もう一つの線分RSについても、  
 R点では、 $Lr = 64 * 3 + 4 = 196 \quad \therefore Mr = 1C4H$   
 S点では、 $Ls = 64 * 3 + 11 = 203 \quad \therefore Ms = 1CBH$   
 となる。  
 よって、

>symdeb

に引き続き、

-f 100 1100 00

で、予め画像データを確保するメモリ領域をすべて0に初期化した後、

-f 184 18B ff

で、MpとMqの両番地間をffで埋め、さらに、

-f 1C4 1CB ff

で、MrとMsの両番地間をffで埋めることにより、命題の横線に対応したイメージ・メモリデータがメモリの中に来る。そして引き続き、

-rcx 1000

-n b:examp.dat

-w

の過程を経て、最後の-wでイメージ・メモリデータをロット・ディスクに格納する。

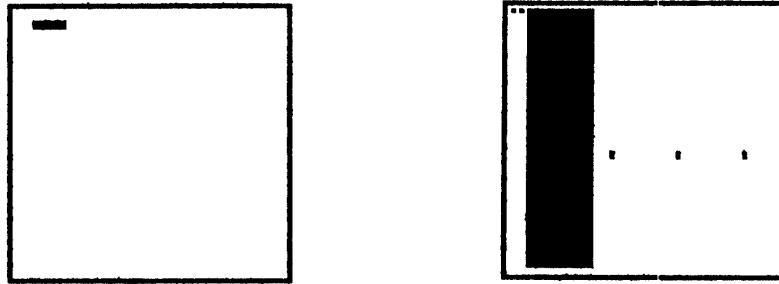
ここで、画像領域の縦横の長さはともに64であるから、 $64*64=4096$ を16進数に直すと1000Hになる。そこで、この数値をrcxによりCXレジスタに設定する。

また、-nにより登録した"examp.dat"は、データに付けたファイル名である。

こうして作ったデータのメモリ構造を(図2)に示す。この状態を、DATA-A型とする。このデータをfread()してdisplay()で画面に表示すると、(図3(a))のように、試験的に作った横線の画像が表示される。display()で表示した画像に対してgetimage()の処理をすると、(図4)のようにデータの並びが完全に变化する。このデータ構造をDATA-B型とする。このデータを利用したdisplay()画像は、(図3(b))のように画像が破壊されている。

0100	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0110	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0120	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0130	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0140	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0150	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0160	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0170	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0180	00 00 00 00 FF FF FF FF FF FF FF 00 00 00 00
0190	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
01A0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
01B0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
01C0	00 00 00 00 FF FF FF FF FF FF FF 00 00 00 00
01D0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
01E0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
01F0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

(図2) EXAMP.DAT (DATA-A形式) の内部メモリ構成



(a) DATA-A型

(b) DATA-B型

(図3) "EXAMP.DAT"のグラフィックス表示

0100	7F 00 7F 00 FF FF FF FF FF FF FF FF FF FF
0110	FF FF FF FF 00 00 00 00 00 00 00 00 00 00
0120	00 00 00 00 00 00 00 00 00 00 00 00 00 00
0130	00 00 00 00 00 00 00 00 00 00 00 00 00 00
0140	00 00 00 00 FF FF FF FF FF FF FF FF FF FF
0150	FF FF FF FF 00 00 00 00 00 00 00 00 00 00
0160	00 00 00 00 00 00 00 00 00 00 00 00 00 00
0170	00 00 00 00 00 00 00 00 00 00 00 00 00 00
0180	00 00 00 00 FF FF FF FF FF FF FF FF FF FF
0190	FF FF FF FF 00 00 00 00 00 00 00 00 00 00
01A0	00 00 00 00 00 00 00 00 00 00 00 00 00 00
01B0	00 00 00 00 00 00 00 00 00 00 00 00 00 00
01C0	00 00 00 00 FF FF FF FF FF FF FF FF FF FF
01D0	FF FF FF FF 00 00 00 00 00 00 00 00 00 00
01E0	00 00 00 00 00 00 00 00 00 00 00 00 00 00
01F0	00 00 00 00 00 00 00 00 00 00 00 00 00 00

(図4) (DATA-B 形式) の内部メモリ構成

ここで、(図2) の原画データの第1, 2行の並びが、

00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00

であるのに対して、(図4) の第1, 2行は、

7F 00 7F 00 FF FF FF FF FF FF FF FF FF FF
FF FF FF FF 00 00 00 00 00 00 00 00 00 00

のように7Fから始まっており、全く理解しがたい数値の並びに変化している。

尚、(図4) のデータに対しては、putimage( ) で正しい画像を表示させることが出来る。

(2) メモリに格納された画像データの状況を調べたい時は、

```
>syndeb b:ファイル名
-d100
```

とする。(図2) 、(図4) は、この方法で確認したものである。

#### 4. あとがき

画像処理の過程で、画像データをフロッピーディスクに保存した場合、その格納の状態に2通りの形式があることが判明した。そして、その2つをDATA-A型とDATA-B型で区別した。DATA-A型は処理方法によってはDATA-B型に変更することがある。

本研究で述べた画像データの作成法は、原図データの書き込みが面倒だが、研究に必要とするデータが自由に出来て便利である。例えば、画像処理の一貫としてアフィン変換や画像の特徴抽出等の基礎研究の対象となる希望の画像データが随意に作成出来る。

#### 参考文献

- (1) 白崎智義：パレット番号を利用した画像輪郭の抽出法の研究，福井工業大学研究紀要，第20号(1990)，51～60
- (2) 白崎智義：画像階調の変換に関する研究 ―面積5階調から8階調へ―，福井工業大学研究紀要，第22号(1992)，63～70

(平成8年12月16日受理)